Utilising the ArchiMate framework to model secure micromobility services

A review of ArchiMate, security modelling and micromobility enterprise

Augustus William Ellerm

Under the supervision of

Miguel Ehécatl Morales-Trujillo

A thesis presented in partial fulfilment for the degree of Master of Science

Computer Science and Software Engineering
The University of Canterbury
New Zealand

April 2021

Utilising the ArchiMate framework to model secure micromobility services

A review of ArchiMate, security modelling and micromobility enterprise

Augustus William Ellerm

Abstract

This research investigates how the EA modelling language ArchiMate can be extended to provide support for security modelling for application in the micromobility enterprise context. EA methodologies enable the governance of large business structures and support the integration processes for IT systems. These benefits align well with the current needs of enterprise security, with stakeholders showing increasing interest in security governance and compliance. Micromobility enterprises are a good candidate for EA security modelling as they utilise quickly emerging and evolving technologies.

To investigate the unique, intersecting, context of ArchiMate, security modelling, and micromobility enterprises; a four phase methodology was defined. First, a scoping SMS study was held, identifying the current state of research and informing the SLR methodology. Second, an SLR study was held, identifying 71 primary papers for synthesis. Third, interviews within industry and government were performed to enrich and contextualise the findings of the SLR. Fourth, the synthesis of the SLR and interviews was completed.

The results of this research consist of four contributions. First, a mapping of current micromobility research is provided through the SMS study. This outlines a current lack of micromobility specific research. Second, an initial identification and classification of micromobility specific security considerations is provided. Third, interviews regarding micromobility, EA and security modelling provide deeper contextualisation of the research and finally, the SLR synthesis, which identifies current issues, gaps in research and general findings.

Preface

The findings presented in Chapter 4 were presented at the SEAA conference 2020 [1]:

A. Ellerm and M. E. E. Morales-Trujillo, "Modelling Security Aspects withArchiMate: A Systematic Mapping Study," SEAA, pp. 577–584, 2020. DOI: 10.1109/seaa51224.2020.00094

Funding

This research was supported by the G B Battersby Trust through the G B Battersby-Trimble Scholarship.

Acknowledgement

Firstly, I would like to thank my supervisor, Miguel Morales, for your unwavering support during the writing of this thesis. Your unrelenting patience and teaching ability provided a consistent leaning environment due to which I can confidently say that I am a better academic and thinker. I have learned about the academic world through you, and due to you, have experienced publishing and actively participating in it. These skills, and many others, will be useful well into the future.

Secondly, I would like to thank the G B Battersby Trust for their generous scholarship opportunities, supporting computer science and software engineering research every year.

Finally, I would like to thank my family, friends, and partner for the support they have offered during this period of my life. Without it, I wouldn't have had the opportunity to pursue this degree, and for that I am forever thankful.

Contents

1 Introduction **14**

2 Background **16**

2.1 Micromobility 16

2.1.1 Micromobility Enterprise 17

2.1.2 Security in Micromobility Enterprise 19

2.2 Enterprise Architecture 23

2.2.1 Visual Modelling Languages 26

2.2.2 Implementing visual modelling in EA 26

2.2.2.1 ArchiMate 27

2.2.3 EA security modelling and Micromobility 32

3 Methodology **33**

3.1 Epistemology and Ontology 33

3.2 Method Design 33

3.2.1 Scoping SMS 33

3.2.2 SLR 35

3.2.3 Interviews 35

3.2.3.1 Interview Protocol 36

3.2.3.2 Interview Structure 36

3.2.3.3 Interview Analysis 37

3.2.3.4 Limitations 37

3.3 Information synthesis 38

4 SMS **39**

4.1 Introduction 39

4.2 Methodology 39

4.2.1 SMS Research Questions 39

4.2.2 Identification of primary papers 40

4.2.3 Data extraction 42

4.3 Results 42

4.3.1 RQ1. How are security aspects being incorporated into ArchiMate EA models? . . . 43
4.3.2 RQ2. What elements are required in ArchiMate to model security aspects in the context of micromobility? . . . 45
4.3.3 RQ3. What security patterns languages do micromobility companies currently employ? . . . 47
4.3.4 RQ4. What support do architectural modelling languages provide for security in EA . . . 48
4.4 Discussion . . . 49
4.5 SMS conclusions and future work . . . 50

5 SLR 52
5.1 SLR Method . . . 52
5.1.1 Planning Phase – Review Protocol . . . 52
5.1.1.1 Research Questions . . . 53
5.1.1.2 Search Strategy . . . 53
5.1.1.3 Study Selection Criteria . . . 56
5.1.1.4 Data Extraction Strategy . . . 57
5.1.1.5 Data Synthesis Strategy . . . 57
5.1.2 Limitations . . . 58
5.2 Results and Analysis . . . 59
5.2.1 Meta analysis . . . 59
5.2.1.1 Publication Year . . . 60
5.2.1.2 Predominant Authors . . . 60
5.2.1.3 Primary Paper Types and Citation Statistics . . . 62
5.2.1.4 Contribution Analysis . . . 62
5.3 Qualitative Synthesis . . . 67
5.3.1 RQ2: What elements are required in ArchiMate to model security aspects in the context of micromobility? . . . 67
5.3.1.1 Topic 1: Attack Vectors and Considerations . . . 67
5.3.1.2 Topic 2: Element Design . . . 82
5.3.1.3 Topic 3: Overall Design Factors . . . 89
5.3.2 RQ3: What security strategies and mitigation techniques do micromobility companies currently employ? . . . 98
5.3.2.1 Topic 1: Privacy . . . 98
5.3.2.2 Topic 2: Security . . . 104
5.3.2.3 Topic 3: Industry Standards . . . 124
5.3.3 RQ1: How are security aspects being incorporated into ArchiMate EA Models? . . . 130
5.3.4 RQ4: What support do architectural design languages provide for security in EA? . . . 133

5.3.4.1 Topic 1: Attack Description . . . 133
5.3.4.2 Topic 2: Risk Assessment (RA) . . . 138
5.3.4.3 Topic 3: Architectural Automation . . . 140
5.3.4.4 Topic 4: Cybersecurity Modelling Languages . . . 143
5.3.4.5 Topic 5: Other . . . 145

6 Discussion 150
6.1 RQ1 . . . 150
6.1.1 ArchiMate's coverage of security applications . . . 150
6.1.2 Incorporating new behaviours into ArchiMate . . . 151
6.1.3 Agile security processes using ArchiMate . . . 152
6.1.4 Conclusions . . . 153
6.2 RQ2 . . . 153
6.2.1 Understanding attack vectors . . . 154
6.2.2 Security Quality Attributes and Application Domains . . . 155
6.2.3 Instantiated Elements and Vulnerable Components . . . 157
6.2.4 Overall Design Considerations . . . 158
6.2.5 Conclusions . . . 159
6.3 RQ3 . . . 160
6.3.1 Security Methods . . . 160
6.3.2 Technical Security and Privacy Solutions . . . 161
6.3.3 Industry Standards . . . 163
6.3.4 Conclusions . . . 164
6.4 RQ4 . . . 165
6.4.1 Representation of Attacks . . . 165
6.4.2 Cybersecurity Modelling Languages . . . 166
6.4.3 Conclusions . . . 167
6.5 Interview Insights . . . 168
6.5.1 Industry perspective . . . 168
6.5.2 Governmental perspective . . . 170
6.5.2.1 Conclusions . . . 171
6.6 Overall Observations . . . 172
6.6.1 Safety as a growing concern . . . 172
6.6.2 A new understanding of Privacy . . . 173
6.6.3 Probability and modelling . . . 174
6.6.4 Automation of architectural descriptions . . . 175
6.6.5 Architectural Frameworks and Structure . . . 176
6.6.6 Architectural Drivers and Cross Cutting Elements . . . 177
6.7 Future Research . . . 179

7 Conclusion **181**
7.1 Future Work 183
7.2 Final thoughts 183

List of Acronyms **184**

A Interview Structure **210**

B Prevalent Authors **212**

C Nvivo Classifications **214**

D Cross Cutting Examples **216**

E Attack Graph Abstraction **220**

List of Figures

2.1 New micromobility platforms per year (data extracted from [5]) . . . 17
2.2 MaaS: Micromobility user timeline . . . 18
2.3 Short trips generate the most emissions (Extracted from [10]) . . . 19
2.4 Connected vehicle risk vectors identified by Deloitte (Extracted from [16]) . . . 22
2.5 EA planning methodology (Adapted from [17]) . . . 23
2.6 The history of EA (Adapted from [19]) . . . 25
2.7 TOGAF and ArchiMate alignment (Extracted from [2]) . . . 28
2.8 Archimate core and full frameworks (Extracted from [2]) . . . 30
2.9 ArchiMate RSO specialisation (extracted from [26]) . . . 31

3.1 Research Method . . . 34

4.1 Distribution of studies per year and type . . . 43

5.1 Classification scheme used to describe RQ2 themes . . . 58
5.2 Publication years of primary papers . . . 60
5.3 Contributions per year per RQ . . . 61
5.4 Predominant Authors and their associated publications . . . 61
5.5 Primary paper type and citation statistics . . . 62
5.6 Contribution percent per RQ . . . 63
5.7 SLR process including snowball extension . . . 65
5.8 CPS Meta-Model (extracted from [70]) . . . 84
5.9 Model of VehicleLang (extracted from [75]) . . . 84
5.10 VehicleLang architecture with proposed privacy extension elements (extracted from [60]) . . . 85
5.11 Combined safety and security view with possible causal chains (adapted from [76]) . . . 86
5.12 Distributed Control Systems Model (adapted from [73]) . . . 88
5.13 Example of MVS modelling (extracted from [79]) . . . 88
5.14 Example of MVS modelling (adapted from [69]) . . . 92
5.15 CloudWoT Framework (adapted from [68]) . . . 93
5.16 Five layered IoT architecture (adapted from [90]) . . . 94

5.17 IIoT to UAF layer mapping (adapted from [89]) 95
5.18 CloudWoT architecture with crosscutting security architectural concern (extracted from [108]) 96
5.19 IoT and CPS consolidated technical architecture. Adapted from [46] . 99
5.20 SafeSecTropos method consolidated from the Secure Tropos and STPA methods. (Adapted from [74]) 106
5.21 Proposed extended risk-based method (extracted from [27]) 108
5.22 Microsoft Threat Modeling Example (Extacted from [68]) 108
5.23 GHOST architecture (extracted from [72]) 110
5.24 S&S model (extracted from [67]) 111
5.25 Associated Collaborative Analysis Framework (extracted from [67]) . 111
5.26 Model Driven Risk Analysis Process (Adapted from [41]) 113
5.27 Proposed RA criteria (extracted from [66]) 115
5.28 Unified Security Patterns (extracted from [42]) 121
5.29 High Level Patterns (extracted from [42]) 121
5.30 Architectural classification scheme based off cyclic groups (adapted from [170]) 132
5.31 SCADA attack tree - potential attacks (adapted from [172]) 134
5.32 Proposed Bayesian/AD Tree Framework (extracted from [175]) 136
5.33 securiCAD aggregation extension (extracted from [176]) 137
5.34 Abstracted attack step of Access Control (extracted from [176]) . . . 137
5.35 Example PRM meta-model (extracted from [179]) 139
5.36 Joint BPM and ERM lifecycles (extracted from [180]) 140
5.37 Example joint utility heuristic (extracted from [184]) 142
5.38 SafeSlice architecture (extracted from [185]) 143
5.39 Proposed EA visualisation approach (extracted from [192]) 146
5.40 Top level ATM functions (extracted from [195]) 148
5.41 The EA analysis process (extracted from [183]) 149

6.1 ArchiMate full framework (extracted from [2]) 178

C.1 Nvivo classifications for RQ1 214
C.2 Nvivo classifications for RQ2 214
C.3 Nvivo classifications for RQ3 215
C.4 Nvivo classifications for RQ4 215

D.1 Cross-cutting security architectural driver in a smart city architecture (extracted from [165]) 217
D.2 Cross-cutting assurance architectural driver example (extracted from [169]) 218
D.3 Motivation aspect from ArchiMate framework (extracted from [2]) . . 219

E.1 Attack graph before abstraction (extracted from [177]) 220
E.2 Attack graph after abstraction (extracted from [177]) 221

List of Tables

2.1 List of micromobility companies (Extracted from [14]) 20

4.1 Search string terms 40
4.2 Classification of keywords 42
4.3 Primary papers associated with the RQs 43

5.1 Search string terms 55
5.2 Highest and lowest contribution by topic 64
5.3 Contribution of each primary paper to each RQ 66
5.4 Attack Vectors identified in the social domain 72
5.5 Attack Vectors identified in the physical domain 72
5.6 Attack Vectors identified in the cyber domain 73
5.7 Security attributes extracted from primaries 79
5.8 Security and Safety objectives defined by [74] 80
5.9 Three layer IoT architectures 90
5.10 Four layer IoT architectures 90
5.11 Five layer IoT architectures 90
5.12 Seven layer IoT architectures 90
5.13 Three layer CPS architectures 91
5.14 Four layer CPS architectures 91
5.15 Five layer CPS architectures 91
5.16 Six layer CPS architectures 91
5.17 Privacy Technical Solutions 102
5.18 Methods of security and safety co-engineering (extracted from [107]) 104
5.19 System actions defined in [71] 107
5.20 Frameworks discussed in [72] 109
5.21 Security Automation Strategy [132] 112
5.22 Information and cybersecurity maturity models (extracted from [133]) 114
5.23 Technical Mitigation techniques by application domain 116
5.24 Technical solutions identified in [64] 118
5.25 Technical solutions described in [65] 122
5.26 Technical solutions described in [69] 122

5.27 Technical solutions described in [62] 122
5.28 Technical solutions described [63] 123
5.29 Referenced Standards 128

B.1 Security and Safety objectives defined by [74] 212

Chapter 1

Introduction

This research explores the duel contexts of Enterprise Architecture (EA) security modelling and the emerging micromobility enterprise markets. With the developments of Internet of Things (IoT) and Cyber Physical Systems (CPS) technologies in the transport domain, two primary shifts can be observed – the establishment of new forms of transport, and the emphasis on the secure and safe application/implementation of these technologies. Part of this development was micromobility vehicles such as electric scooters.

With the advent of micromobility vehicles came the wave of micromobility platforms and industry, providing transport in urban centres through connecting consumers with vehicles via smart phone applications. Enterprises offering Transport as a Service (TaaS), micromobility enterprise in particular, are exposed, and expose to the public, a unique set of security and physical risk. Physical tampering of provided vehicles, hacking and exploitation of application software, hacking of the vehicle itself, and privacy concerns are primary considerations for this growing industry. Due to these unique risks micromobility enterprises need to provision security as a primary aspect of their operation.

Enterprise describes an organisation or set of organisations which operate and collaborate to provide a service or product to consumers. EA is one method which has been leveraged in the past to provide governance of various aspects of an enterprises operation. Business process design, infrastructure design, development plans, documentation, and identification of improvement opportunities are just some of the benefits that enterprise architects enable through EA. Specific tooling has been developed alongside these methods in order to support their application. One primary tool used by architects is modelling, in which aspects of the process are described – either visually or formally. ArchiMate is a well regarded and widely cited architecture modelling tool which is developed by The Open Group (TOG) for use alongside their architecture framework and specialises in producing models which describe, analyse, and communicate concerns of EA as they evolve over time [2].

ArchiMate, however, does not provide security as a primary concern within its

modelling framework making it difficult – if not impossible – to implement integrated descriptions of security solutions, risk and possible attack vectors within its overarching model. In the context of connected transport security governance and development is a primary consideration for stakeholders within the enterprise. EA and EA modelling is already poised to deliver value in regards to other aspects of enterprise development at this level, so how can it be leveraged in the security domain?

As well as answering this question, this research contributes as a first step in the definition of the micromobility security landscape – an emerging conversation in the wider context of connected and autonomous vehicles (AV's). Micromobility offers a unique instantiation of IoT/CPS/connected vehicle technologies due to its explosive introduction and widespread utilisation in the last five years. The fast introduction and implementation of these technologies, while beneficial in many aspects, has also lead to security failures and dangerous scenarios. By utilising EA modelling to provide security governance, these enterprises could perform agile adjustments to their security posture and processes in a timely and cost efficient way, while identifying security as a primary concern to their stakeholders.

This outlines the primary objective of this research – to identify how EA modelling frameworks can be utilised to provide value to micromobility security concerns. To achieve this objective three contributions are expected to be made. First, a consolidation of security in micromobility. Second, An understanding of current methods of security modelling in both the EA context and wider modelling contexts. Finally, an identification of the current EA security modelling solutions, their feasibility in the micromobility context, and the identification of improvement opportunities.

This research is structured into seven chapters. Chapter two presents the history and background on the micromobility and EA concepts as well as framing the research area. Chapter three presents the overall methodology followed during this research and their justifications. Chapter four presents the results of the Systematic Mapping Study (SMS). Chapter five presents the results of the Systematic Literature Review (SLR). Finally, Chapter six provides a discussion on the researches findings and Chapter seven presents the researches conclusion.

Chapter 2

Background

This chapter presents the problem space the research is investigating, describing the primary topics of Micromobility and EA. A history and description of these topics is provided, enabling greater understanding of the motivation behind this topic – how security, in the context of micromobility can be effectively and intuitively integrated into ArchiMate.

2.1 Micromobility

The literal definition of micromobility describes travel (mobility) over a short distance (micro). Recently this word has been extended to the vehicle domain, denoting a class of vehicle – supposedly first termed by Horace Dediu in 2017, a Romanian-American industry analyst [3]. Micromobility, the class of vehicle, describes small electric vehicles (EV's) weighing below 500kg which implement some form of short distance travel.

The growth in small connected EV's has seen a steady increase in popularity with governments and foundations, such as the Society of Automotive Engineers [4] (SAE), providing further categorisation under the umbrella of micromobility. Along side this, Micromobility Industries has provided a more detailed categorisation with three characteristics that all micromobility vehicles fulfil. First, the vehicle must weigh less than 500kg; Second the vehicle must be electric; and finally, the vehicle must be utility focused [5]. Common vehicles conforming to this definition include electric scooters, electric bikes, hover-boards and segways.

The definitions that micromobility institutions use, however, have been unable to agree on one factor – whether or not micromobility vehicles are required to be motorised (electric). On this topic Horace Dediu has written:

> "This is the interesting thing, of course: we've had bicycles for a long time. But what micromobility is, is a motorised vehicle. Earlier I said micromobility is 500kg or less but I would amend that definition to

include (a) having a motor and (b) being used for utility transport." [6]

In this way micromobility is not only a class of transport, but also a descriptive term for the transformation happening within the transport domain. Some governmental bodies and research agencies have defined micromobility as any form of transport which weighs less than half a tonne. This would include, for example, push bikes, skateboards and even roller-skates. This research uses the former definition, including vehicles which are in some way enabled by an electric motor. The term non-electric micromobility will be used to reference vehicles which are not enabled by an electric motor.

Historically non-electric micromobility services were proposed for the first time in Europe in 1975 in the form of a community bicycle program by Luud Schimmelpennink [7]. Electric micromobility was first available to the consumer in 1996; however, the wide adoption as seen today can be attributed to micromobility enterprise's – the first of which begun operations in late 2017.

2.1.1 Micromobility Enterprise

Micromobility enterprise has encouraged the wide spread adoption of micromobility vehicles, providing mobility as a service (MaaS) to cities around the world. Businesses that provide these services are popping up around the globe (see Figure 2.1), with many following Bird's – a first mover and industry leader in micromobility enterprise – lead in providing e-scooters to the public for rent.

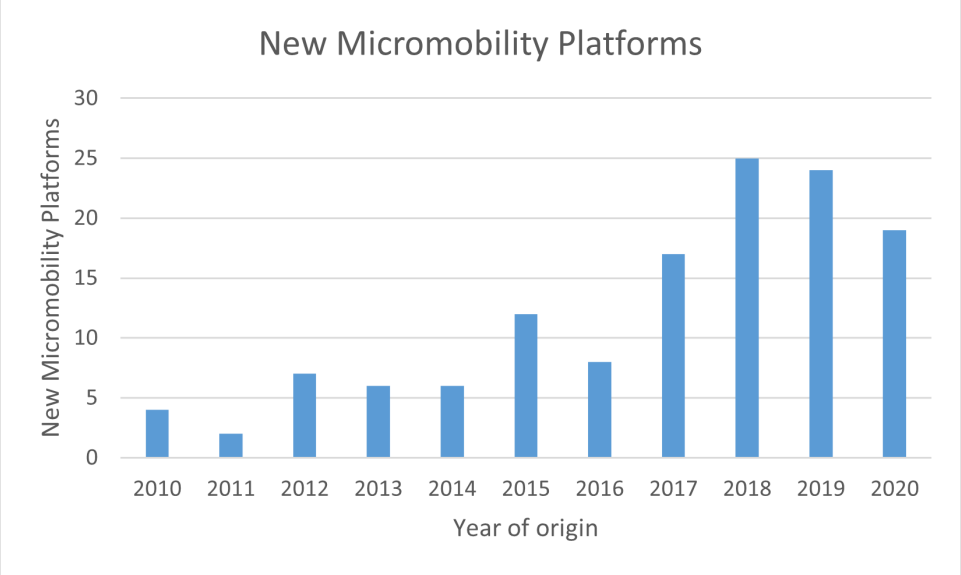


Figure 2.1: New micromobility platforms per year (data extracted from [5])

MaaS business strategies integrate all stages of traditional travel – trip planning, booking, ticketing and payments into one experience. A popular method of achieving

this integration is through the mixture of smartphone applications and physical devices in the public domain. Micromobility enterprises follow this strategy, providing an app to consumers who can plan their trip, locate a vehicle, rent that vehicle and arrive at their destination (see Figure 2.2).

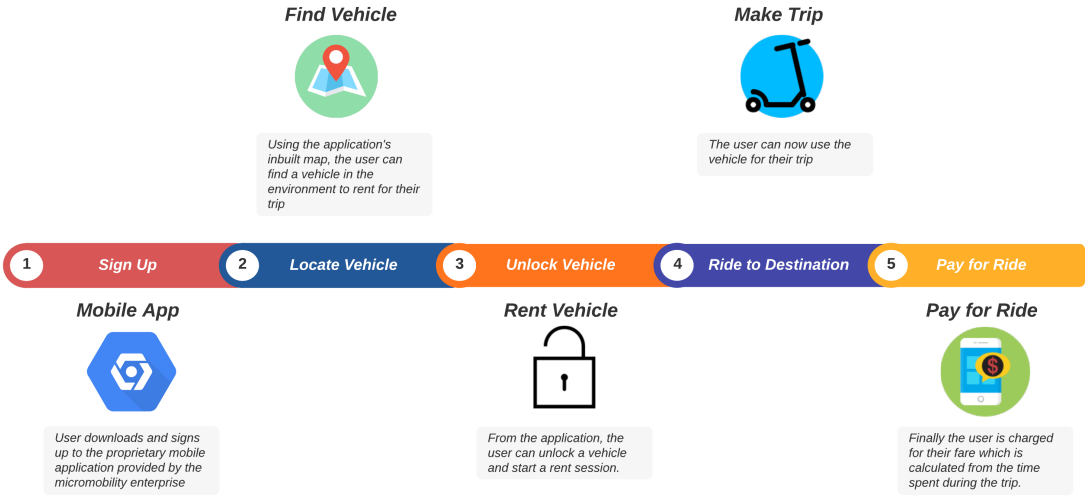


Figure 2.2: MaaS: Micromobility user timeline

The popularity of micromobility enterprise can be attributed to the increasing desirability of short-distance transport in urban centres. Urban centres are becoming increasingly congested, creating unpredictable travel times at peak hours and other negative effects. Residents of Los Angeles, Sydney and Singapore spent an extra 25, 19 and 18 minutes respectively in traffic during rush hour [8] [1]. In New Zealand, Auckland residents spent 20 minutes longer travelling by car during congested times – making micromobilities an attractive commute option for those who live there. Other factors, such as cost and accessibility have also been cited in relation to micromobilities new found popularity [9].

Other than benefits to the public, there are larger scale benefits that governmental bodies are seeking from micromobility transport. For example, accessibility of transport options for low-socioeconomic groups, reduction of pollution due to transport and integration of last mile alternatives into the public transport system.

Perhaps the most widely cited benefit of micromobility transport is its environmental impact. Micromobilities, by definition run off electricity, so in this sense they are more environmentally friendly than vehicles which run off fuel, assuming an appropriate portion of the electricity generated for the vehicle was generated through sustainable means. This low emission running cost, coupled with the fact that short trips in traditional vehicles generate the most emissions (see Figure 2.3) [10] has poised micromobility to make substantial gains in this area.

[1] Statistics taken from 2019 due to COVID-19's effect on 2020 data

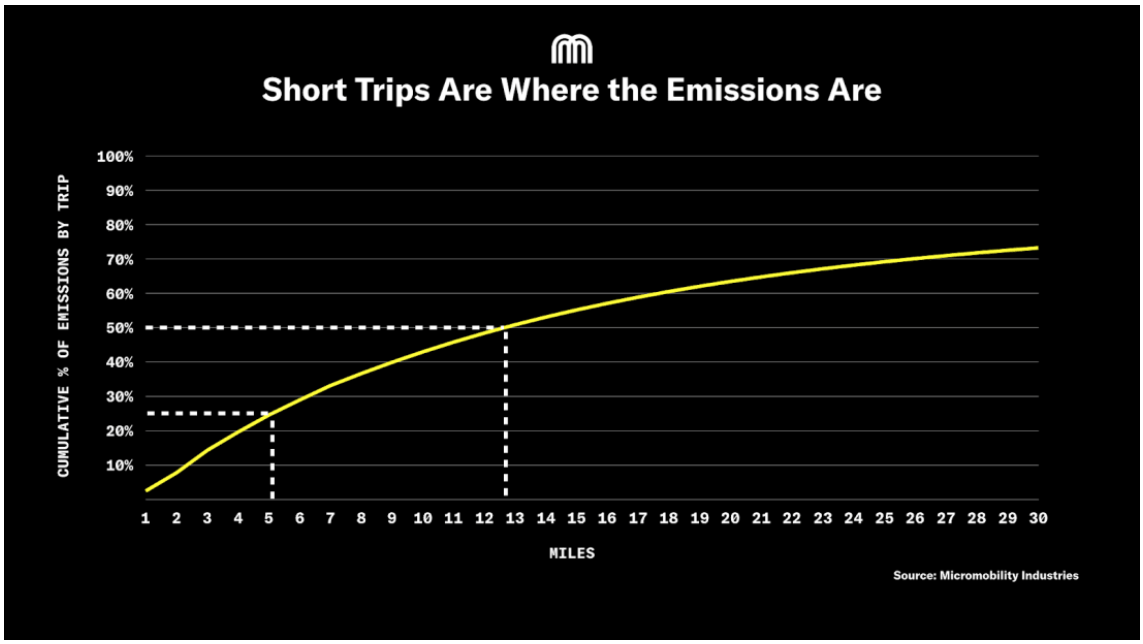


Figure 2.3: Short trips generate the most emissions (Extracted from [10])

Critics, however, have pointed to the short lifecycle that micromobility vehicles often exhibit. Due to vandalism, theft and general environmental factors e-scooters in August 2018 were shown to have an average lifespan of only 28.8 days [11]. Not only does this produce more waste – and therefore more pollution – but it also affects the profitability of the enterprise.

The poor lifespan of scooters is generally attributed to their design. Initial scooters deployed by the micromobility enterprise Bird were re-branded Xiaomi devices intended for use on flat surfaces, in good weather and with a weight limit of 200 pounds [12]. These factors cannot be accounted for when running a business which deploys these vehicles to anyone and everyone – potentially shortening the devices lifespan. With the rise of micromobility enterprise however comes the rise of proprietary supply chains. Enterprises such as Neuron [13] are selling vehicles like the Neuron N3 e-scooter which are designed with the scooter sharing economy in mind. With these adjustments, and the recycling of damaged vehicles, the lifecycle of micromobility vehicles will increase.

An overview of micromobility enterprises, and their supported transport technology is provided in Figure 2.1. This data is based upon the green paper provided by Ramboll [14] and contains enterprises operating electric and non-electric micromobility enterprises running around the world in 13 different cities.

2.1.2 Security in Micromobility Enterprise

With the advent of micromobility, the MaaS industry boomed with new and innovative ways to provide transport to the public. Taking advantage of IoT technologies

Table 2.1: List of micromobility companies (Extracted from [14])

	e-scooter	e-bike	other (electric)	pedal-bike
beam	✓			
Bolt	✓			
Circ	✓			
DIVY		✓		✓
DONKEY REPUBLIC				✓
EASYBIKE				✓
Flamingo	✓			
grüv	✓			
(hoop)	✓			
JUMP	✓	✓		
Lime	✓	✓		✓
lyft	✓	✓		✓
mobike				✓
neuron	✓			
nextbike				✓
onzo				✓
(Oslo Bysykkel)				✓
samocat	✓			
SHERPA	✓			
SPIN	✓	✓		
StadtRAD Hamburg		✓		✓
TIER	✓			
VeoRide	✓		✓	
voi.	✓	✓		
WHEELS			✓	

and cellular networks, each individual vehicle in an enterprise can be tracked through Global Positioning System (GPS) data. This substantially lowers the risk of theft and misuse through enabling non-repudiation. While the benefits afforded by connected vehicles are great, with this added complexity comes security and safety considerations.

IoT vehicles and micromobility vehicles contain tightly coupled security and safety considerations due to their ability to cause physical harm to a user or a user's privacy. This is due to governance mechanics built into these devices in order to lower the risk to an enterprise, effectively enabling the operation of MaaS business models. Two primary governance benefits enabled by IoT/CPS technologies are the ability to track vehicles through GPS data, and the remote access and/or control of vehicular systems. These systems provide sufficient governance of an enterprise's investment to warrant providing expensive vehicles in the public domain – were tampering and misuse are probable. These techniques, however, also introduce security concerns tightly coupled to safety objectives. For example, the ability to remotely control aspects of the vehicle requires an interface/API to be implemented, which exposes the risk that this functionality is misused. A direct consequence of this is bodily harm, which is represented as a safety concern. In this way, the coupling of security and safety in the connected vehicles domain has grown in strength.

Micromobility enterprise, specifically, must be aware of security and safety as licensing and operation agreements offered by governmental authorities, are dependent on the perception of security and safety of the enterprise's product. In this sense, the business' profitability relies on the perception of secure and safe systems.

Micromobility enterprises find themselves in a place of responsibility due to the service they provide and must maintain high security and safety standards in order to operate in a high risk context. This task is made harder due to the disruptive nature of micromobility vehicles – infrastructure was not, and has not been provided specifically for micromobility transport. There are no cycle-ways for micromobility vehicles and the transport policy regarding how and where they should be utilised is lacklustre. Collaborating with government transport departments to minimise these risks should be considered as without proper precautions and policy in place the view of micromobility may sour in the future.

A quick Google search of "e-scooter hack" will provide much evidence that the security in regard to these scooters has been lacking. Exploits that range from removal of a top-speed limiter to full theft and modification into personal devices can be found on widely available networks such as YouTube, who seem to be ambivalent regarding whether or not these videos are allowed on its platform. These exploits, however, are comparable to 'script kiddies' - that is, low skill exploits which are inflexible. Due to the inherent inflexibility of these attacks it proves relatively easy

to mitigate. More dangerous, in-depth exploits have been published such as the application released by Zimperium, a mobile security company. The Zimperium blog provides more context on the critical security failure:

> "During our research, we determined the password is not being used properly as part of the authentication process with the scooter and that all commands can be executed without the password. The password is only validated on the application side, but the scooter itself does not keep track of the authentication state." [15]

This exploit enables an array of attack scenarios, the most dangerous being the ability to accelerate, or suddenly brake without the consent of the user. These attacks were performed on a brand of personal scooter – the Xiaomi M365. This is the same scooter that was repackaged for MaaS deployment, distributing vulnerable units across the world.

For micromobility enterprise to be a sustainable business model in the future, the security and safety of these systems needs to be demonstrable, warranting trust from the public and governmental entities. An overview of the risk vectors associated with connected vehicles is provided in Figure 2.4. Micromobility vehicles, as an instantiation of connected vehicles, consist of similar vectors, however, the infotainment systems and advanced/AV aspects are unlikely to be instantiated in micromobility vehicles.

Figure 2.4: Connected vehicle risk vectors identified by Deloitte (Extracted from [16])

2.2 Enterprise Architecture

In 1987, John Zachman published the Zachman Framework, which has been cited as the historical beginning of EA. From this point, EA gained popularity for its ability to continuously integrate information systems into business processes, providing strong governance over an enterprises business strategy.

In 1992, the first proposed EA planing methodology was published – providing a method to create an instantiation of a target EA [17]. This method is abbreviated into five steps provided in Figure 2.5 below [17].

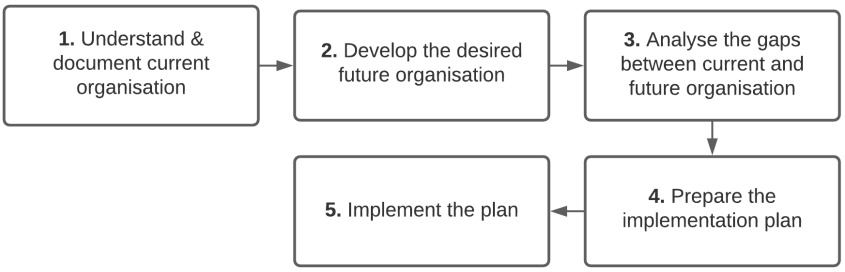


Figure 2.5: EA planning methodology (Adapted from [17])

These steps define the "AS-IS" architecture and the "TO-BE" architectures, identifying the transitional steps required to move from the fist state to the next. EA saw a wide spread adoption past this point, with the American Federal Government, through the Clinger-Cohen Act, requiring all its departments to develop consistent architectures compatible with the National Institute of Standards and Technology (NIST) EA model [18].

Modern EA practice has enabled a new role in the enterprise – the architect. The architect is tasked with maintaining the organisation's Information Technology (IT) networks and services, and aligning them to business goals. This process involves evaluating the current business structure, through metric and information gathering, and identifying how the current "AS-IS" architecture needs to evolve in order to support new business goals, aligning with industry best practice, providing process optimisation and many other desirable objectives. During this process a model of the current business is formed, which can also provide a means of evaluating efficiency, security considerations, cost-effectiveness and other useful indirect metrics.

Over the years, industry professionals and academics have proposed EA frameworks and methodologies, providing tools for architects. These tools help with standardisation, as well as quality control – identifying areas of enterprise which architects should focus on as well as methods of data collection and application. There are several EA methodologies and frameworks available to enterprise architects to-

day, with the most popular being TOG Architectural Framework (TOGAF), The Zachman Framework and the Federal EA Framework (FEAF).

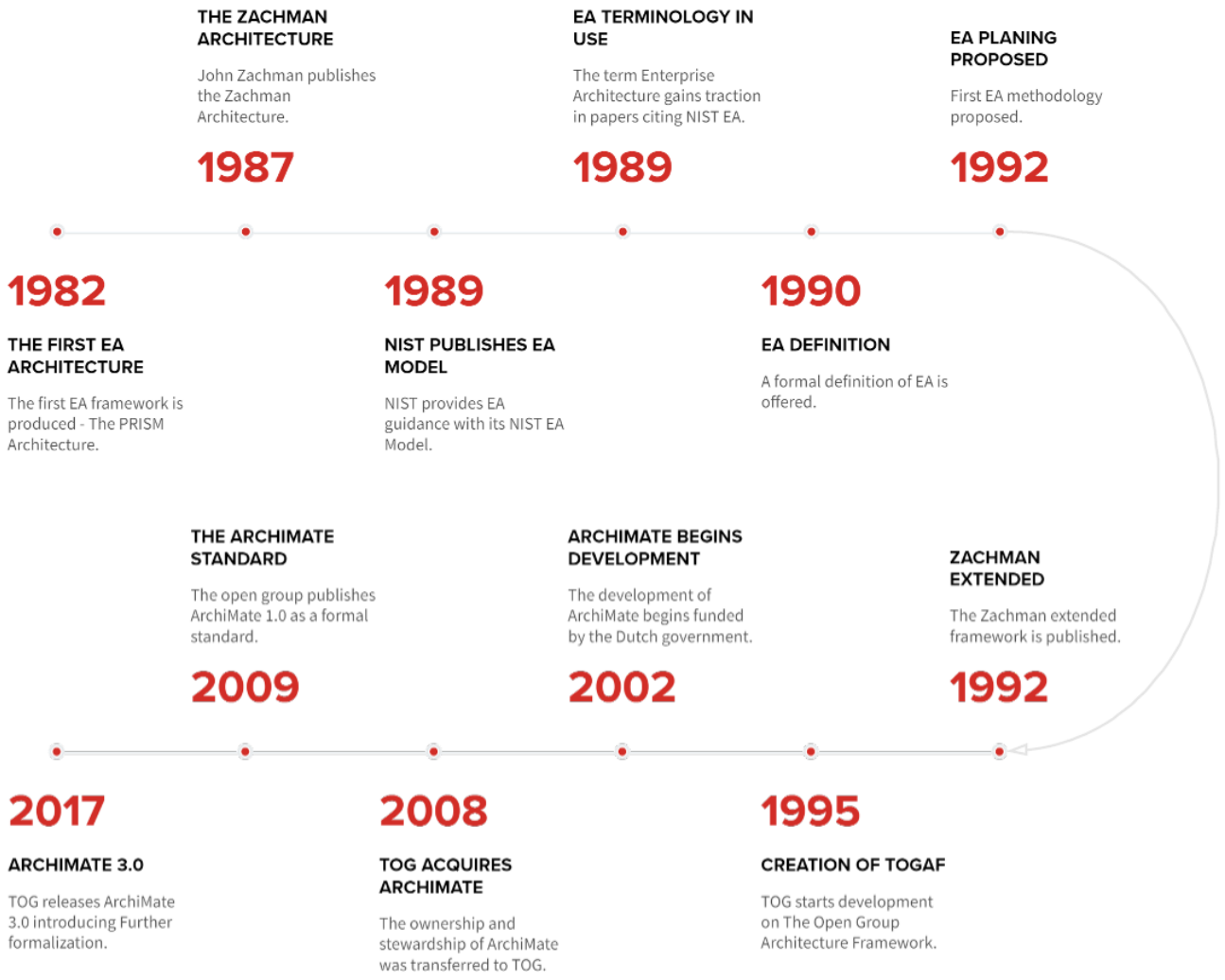


Figure 2.6: The history of EA (Adapted from [19])

2.2.1 Visual Modelling Languages

Visual modelling is a well documented and utilised information communication format and is leveraged in many domains, including information visualisation, geographic information systems, software architecture, building architecture, among others. Visual models provide a variety of benefits such as the ability to remove or reduce complexity using abstraction, increase the understanding of the depicted system through *at a glance* analysis by presenting information with low cognitive overhead [20] and using intuitive modelling methods to provide understanding without expert training.

Hall et al. [21] categorises visual modelling into three sets: Journalistic visualisation, Scientific visualisation, and Critical visualisation. Journalistic visualisation leads the narrative of the visualisation, *telling a story* of the data and leading the reader to a predefined conclusion. It simplifies and explains complex processes and lends itself to interactive mediums, such as interactive geolocation heat maps. Scientific visualisation do not frame the data or processes they are representing and provide an objective visualisation. As such they facilitate interdisciplinary communication between scientific communities, identifying a scientific principle or system. Finally, critical visualisations enable user-lead enquiry and analysis. These methods provide the tools and processes through which an user can enquire into aspects of the visualisation.

Visual modelling languages have been researched and used extensively in software modelling and constitutes a form of critical and scientific visualisation (depending on the context) [22]. Visual models help ease understanding of complex systems, explore and analyse design alternatives, provide an implementation road map, capture requirements, and communicate design decisions [23]. These benefits have been leveraged in the EA method, providing critical visualisations and documentation of EA systems. EA models facilitate analysis, communication and evolution strategy between an architect and stakeholders. Visual modelling provides an unambiguous specification of an EA *AS-IS* state which can then be used to elicit the next iteration of the specification (the *TO-BE* state) and, in doing so, provide the road map consisting of the required amendments to reach the *TO-BE* state.

2.2.2 Implementing visual modelling in EA

Visual modelling has been established as a valuable method of analysis and communication in many contexts, and therefore there are many different methods of implementing visual modelling. Yashchyshyn [24] provides three proposed frameworks for modelling complex and heterogeneous systems.

The three proposed critical visualisation frameworks are the *multi-view modelling*, *amorphous heterogeneity modelling*, and *hierarchical multi-modelling* frameworks [24]. *Multi-view modelling* proposes a set of distinct separate models which represent different areas of the system which, when combined, provide a full description of the modelled system. To combine these models, a formal structure of the model is defined, creating a structured and compartmentalised holistic model. *Amorphous heterogeneity modelling*, similarly to *multi-view modelling* provides many models, however, these are combined in an arbitrary way without regard to structure. Finally, *hierarchical multi-modelling* combines multiple distinct modelling methods using a hierarchical structure in order to take advantage of certain unique characteristics at each level of the hierarchy.

Regardless of the chosen modelling framework, the specificity and generality of the elements proposed within the framework is an important factor in dealing with complexity. For example, composing a modelling language with general, unspecified elements provides the flexibility to deal with many modelling contexts. This approach, however, can be prone to misrepresentation due to its lack of standardisation and descriptive power. The opposite approach – providing specific elements and relationships – improves understandability and descriptive power but can be cumbersome and difficult for the layman to understand [24].

EA visual modelling is traditionally achieved through *multi-view modelling* due to the layer conceptualisation of business processes and EA in standards such as TOGAF, the Zachman Framework and the FEAF. ArchiMate is the primary EA modelling language used today and is provided by TOGAF for use alongside their architecture framework. Due to ArchiMates popularity it was chosen as the subject of this research.

2.2.2.1 ArchiMate

ArchiMate was first proposed by a project team of Telematica Instituut in 2002 and was acquired by TOG who re-released the modelling tool as ArchiMate 1.0 in 2009 (Figure 2.6). ArchiMate is specifically designed to be used alongside TOGAF, providing models and documentation supporting EAs. TOGAF is an enterprise framework which describes methods and approaches for designing, planing, implementing, and governing EA. The alignment of TOGAF and ArchiMate is shown in Figure 2.7.

While this alignment is not a one-to-one mapping of ArchiMate layers to TOGAF processes, it does describe the general alignment between these concepts.

2.2.2.1.1 ArchiMate structure and elements

ArchiMate can be described at two levels of complexity. First, the ArchiMate core framework which defines the concepts and relationships required to model EA. This

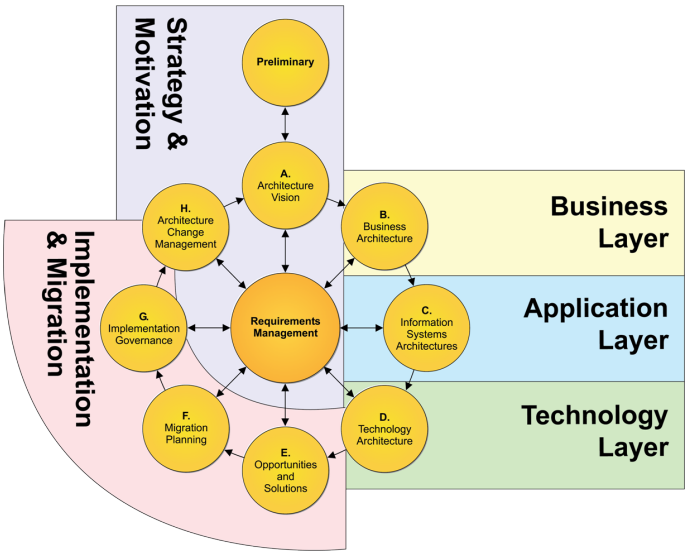


Figure 2.7: TOGAF and ArchiMate alignment (Extracted from [2])

framework consists of three layers – Business, Application, and Technology, and three cross cutting elements – Passive structure, Behaviour, and Active structure (Figure 2.8a).

The *Technology Layer* identifies technology services (processing, sensing, storage, servers, etc.) that support higher level applications. The *Application Layer*, enabled by the technology layer, describes internal application services that support the business. Finally, the *Business Layer* identifies the services and products offered to consumers.

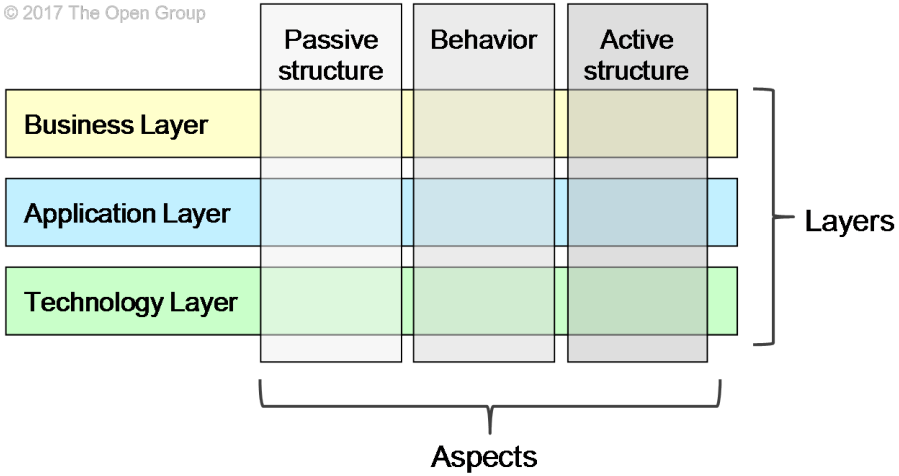
The three cross-cutting elements (Active structure, Behaviour and, Passive Structure) are classified as aspects. The *Active Structure* classifies structural elements and processes which display actual behaviours (i.e. they are not acted upon, but act). The *Behaviour Aspect* classifies structural elements which constitute the actions (or behaviours) performed by actors classified in the *Active Structure*. Finally the *Passive Structure* classifies structural elements on which Actors classified in the *Active Structure* perform behaviours upon [2].

The full ArchiMate framework provides three additional layers and one additional aspect (Figure 2.8b). The layers are *Strategy*, *Physical* and *Implementation & Migration*. The *Strategy Layer* identifies capability, resources and course of action that make up the overall strategies of the enterprise, supporting strategy execution. The *Physical Layer* extends the *Technology Layer* providing a model which describes the physical world - manufacturing, logistics and technology operating environments. Finally, the *Implementation & Migration Layer* relates elements (programs, projects, and processes) to architecture which they implement, supporting migration planning [2].

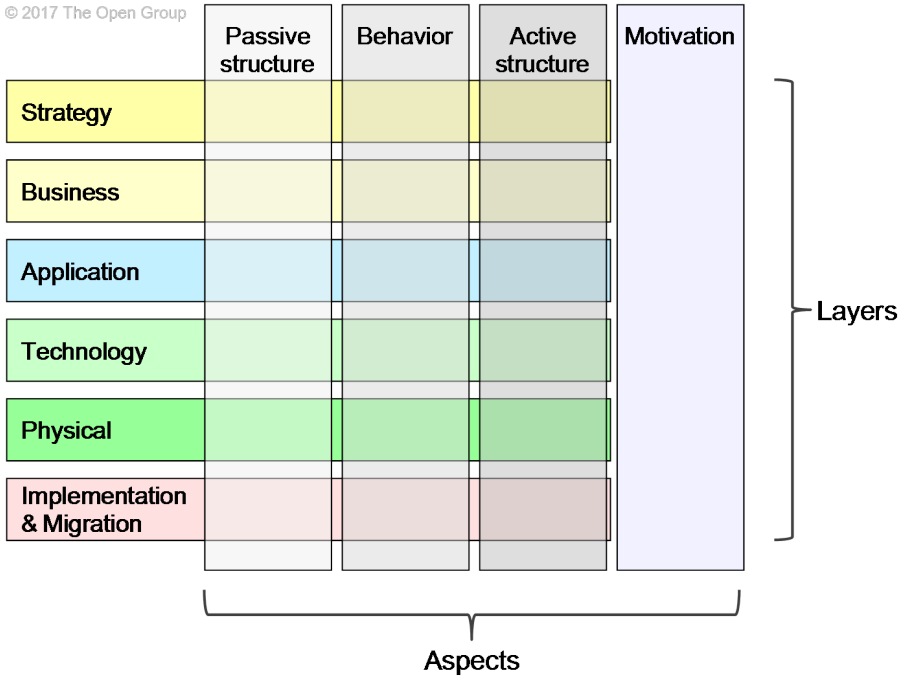
The additional aspect, *Motivation*, provides information on the motivational as-

pects of each process/element in the architecture, providing traceability between the underlying goals and reasoning behind its implementation.

In terms of the distinction of generalizable elements vs. specified elements discussed in Section 2.2.2, ArchiMate tends to provide general elements to allow flexibility in its application.



(a) ArchiMate core framework



(b) ArchiMate full framework

Figure 2.8: Archimate core and full frameworks (Extracted from [2])

2.2.2.1.2 Security modelling in ArchiMate

The Risk and Security Overlay (RSO) extension was proposed for ArchiMate 2.0 and constituted the first direct component of ArchiMate to support security modelling in EA. The RSO was developed in collaboration with security forums and TOG, based on reviews of popular risk frameworks (TOGAF security guide, Sherwood Applied Business Security Architecture (SABSA)) [25]. Using existing language elements, the RSO specialises the element *Business Event* (from the business layer) into two concepts; Threat Event and Loss Event. Elements from the Motivation layer are also specialised into vulnerability, risk, control objective and control measure (see Figure 2.9). These specialisations provide a notation and standardisation for risk and security modelling in EA using ArchiMate.

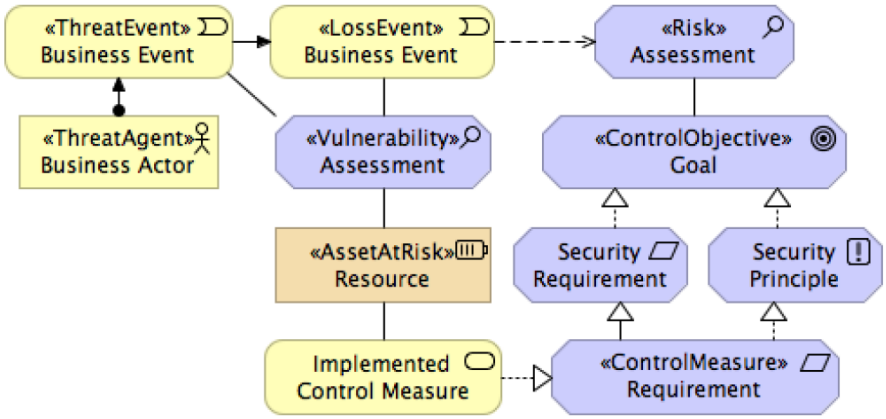


Figure 2.9: ArchiMate RSO specialisation (extracted from [26])

The RSO has received criticism from academics who outline areas in which traditional risk management frameworks out perform it. Prince et al. [26] identifies lacking areas by comparing the ontology of the RSO with the "common ontology of value and risk". The authors propose a redesign which includes new elements to describe concepts such as threat capabilities, vulnerabilities of threat enablers and assets at risk. One reason for why the RSO does not perform as well as these other methods may be to do with TOG's approach to ArchiMate. ArchiMate is a lightweight modelling language, containing relatively few elements. This lends itself to simplicity, making the generated model easier for non-experts to understand. Due to this simplicity, some of the required complexity in risk modelling may be lost. The balance between simplicity and complexity is a distinguishing characteristic of different modelling languages. For example, UML is complex and is afforded greater flexibility for modelling complex processes. This, however, comes at the cost of clarity – the relative opposite problem of ArchiMate.

Another criticism is the lack of a security aspect or layer in the ArchiMate framework. This delegates RSO models to self-contained sub-models which are not integrated into the holistic enterprise model. EA modelling methods need to be

developed which integrate security modelling (risk and literal) into the holistic enterprise model, providing governance and evolution of these systems as an enterprise grows.

2.2.3 EA security modelling and Micromobility

Micromobility enterprises, and MaaS enterprises generally, would benefit through utilising EA methods due to their quickly evolving business models, complex supply chain, third-party service integration, and complex governmental interactions. These enterprises, however, are required to place more emphasis on the security of their systems due to the safety implications of their service. EA modelling techniques, while positioned to provide value in describing enterprise structure do not provide sufficient methods of security modelling.

In EA models, security is often excluded from the wider description of EA and regulated to self-contained models which do not facilitate the governance benefits found within the primary holistic model. With the advent of MaaS and IoT/CPS technologies, security has become a primary goal within these industries. This is confirmed by observations within industry, which identified a growing interest from stakeholders regarding governance of security processes [27]. EA methods need to reflect this growing need and one way of doing so is extending EA modelling into the security domain.

Providing a method of integrating security modelling into the EA methodology would enable critical systems to be acknowledged, governance over the integration, development, and implementation of new security measures while fitting within the objectives of EA - aligning IT and business concerns.

Chapter 3

Methodology

3.1 Epistemology and Ontology

The body of this research is based upon the theory of organisational knowledge creation which constitutes a two-dimensional model of codification and abstraction of knowledge. Using this theory, the method of combination – the creation of new explicit knowledge from found explicit knowledge – is used during the synthesis of information found within the SLR. Further, tacit knowledge is extracted from the relevant domain through an interview methodology which, by using the method of externalisation, is converted to explicit knowledge during the discussion section. These concepts, and their underlying principles are given in [28].

3.2 Method Design

The research method followed in this research consisted of four phases. Phase one involved a SMS study designed to identify the scope and appropriate configuration of the larger SLR (Phase two), providing confidence in the correctness of the SLR's approach. Phase two used the findings of phase one to design a SLR study based upon methods offered in [29] which provided the main outcome of this research. Phase three supplements these findings with real world perspectives through interviews with industry and government experts. Finally, the findings from these phases were used to identify and analyse the research problem, providing insight on the solution space and defining the unique security requirements of micromobility enterprises. The overall method design is described in Figure 3.1

3.2.1 Scoping SMS

As micromobility enterprise and security modelling in EA are both emerging topics a scoping SMS study was designed and performed to provide context on the current state of research in these domains. The findings of this preliminary study

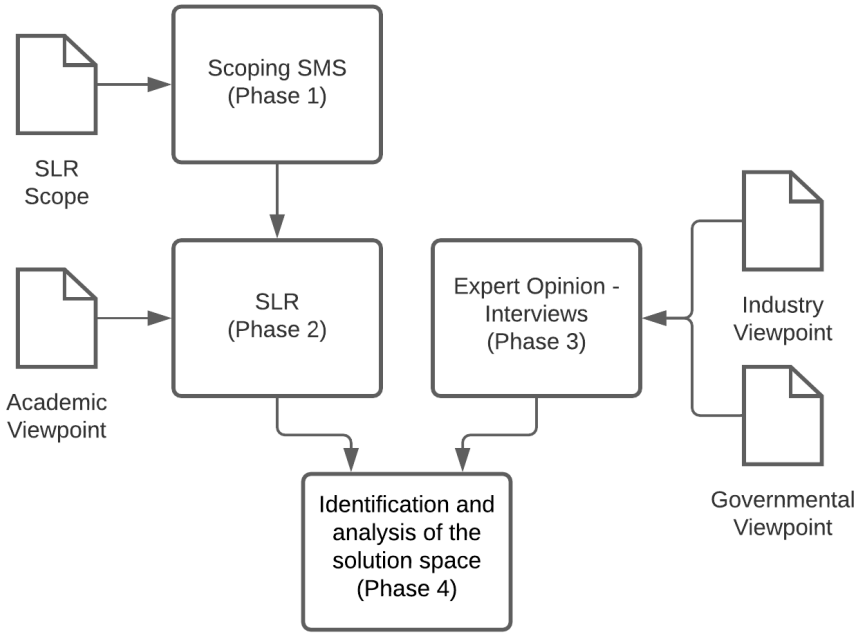


Figure 3.1: Research Method

align the SLR to current research, providing the information necessary to mitigate threats to its validity, such as potentially dismissing relevant research. The SMS also considerably improved the quality of the search strategy used.

The SMS methodology was chosen due to its flexibility, relative overhead and its well documented and defined method [30]. Unlike the SLR method, SMS's are primarily concerned with structuring a research area. This provides a good balance between exploring the security modelling and micromobility research areas and providing actionable information for use in the SLR. Lower overheads in terms of quality assessment, research rigour and the lack of synthesis - which is required in an SLR [30], provides a method which can be completed in a timely manner while achieving the required goals.

The SMS method was based on robust procedures defined in [30], providing confidence in the procedures results and minimising the risk of researcher biases. The SMS procedure itself is similar to an SLR's procedure and requires search terms and a search query to be defined. This query is executed on pre-selected databases which return a set of research documents for further analysis. Through this method research biases found within traditional thematic reviews are minimised and the full range of research relating to the topics can be investigated. A further, in-depth description of the SMS methodology is provided in the SMS chapter (Chapter 4).

3.2.2 SLR

A hybrid SLR method (a method extending the traditional SLR methodology) was selected for this research as it allows research from many domains to be collated and synthesised together in order to provide an answer to the research question (RQ). As this research spans many distinct domains (transport, connected technology, EA, and modelling) the flexibility and completeness enabled by an SLR is beneficial.

SLR's can be published as stand-alone contributions to their chosen field and can help others understand a domain. They deliver concise overviews of the current literature/evidence on a topic and through doing so can identify gaps in research, providing valuable future research opportunities. No SLR (to the knowledge of the author) has explored the emerging context of micromobility and EA, further consolidating the contribution such a work would bring. The chosen SLR method follows procedures defined in [29], and minimises researcher and other biases to provide a clear review of research related to the RQ. Different to SMS methods, SLR's provide a synthesis of information gathered regarding the topic, providing a method to identify and analyse the possible solution spaces for the RQ - one of the primary goals of this thesis.

To extend this SLR and provide the most complete review, the forward snowballing method was implemented. Through utilising citation tracking provided by Scopus, a wider array of research is able to be reviewed. This, in conjunction with traditional SLR methodology (database searches), enables a complete and exhaustive review of the literature. This additional process mitigates traditional limitations of the SLR method, providing another avenue of research identification and enabling greater recall of relevant research [31]. A further, in-depth description of the SLR methodology is provided in the SLR chapter (Chapter 5).

Literature scope is defined by both recall – how much research is returned – and precision – how applicable the returned research is. Three databases were selected for the initial stage of the SLR (IEEE Xplore, Scopus, ACM DL) providing a large recall of research. This is then followed by the snowballing mechanism with utilises one database (Scopus) which has consistently provided the best precision, and recall out of any database index [31]. Using this configuration good coverage of all relevant research is achieved while also acknowledging diminishing returns when over sampling from multiple databases.

3.2.3 Interviews

Interviews were selected as a method to enrich, corroborate and support the findings of the SLR during the identification and analysis of the solution space. Interviews provide valuable insight into industry and government, providing real world feedback on currently utilised solutions and needs. During the design of this research's

methodology, potential interview limitations such as low attendance were taken into account due to the instability caused by the COVID-19 outbreak. To mitigate these limitations more emphasis was placed upon the SMS and SLR which methods do not rely on inconsistent factors.

3.2.3.1 Interview Protocol

Potential interviewees comprised of micromobility enterprises who operated within New Zealand, academics who researched related subjects (security, transport, modelling) and government departments who govern micromobility interactions within urban centres. Contact information regarding these groups was identified through either front facing interfaces (e.g. public emails) or was provided by contacts within academia or industry.

After an expression of interest by the interviewees, an information and consent form was distributed for signing and returning outlining the interview process, expected outcomes and information disclosure policy. All interviewees had the right to withdraw from the interview at anytime, as well as the right to edit the transcription of their interview in order to remove any sensitive information.

The interviews took place over the platform Zoom due to location and public health circumstances. A recording of the interview was taken, and an automated transcription was generated which was edited for readability and correctness. This transcription was used for confirmation with the interviewees regarding the appropriateness of the data to be used, and for analysis purposes.

Finally, Excel was used to classify and breakdown each interview into useful themes and observations on the research topic reported in Section 6.5.

3.2.3.2 Interview Structure

Due to the diversity of interviewees and discussion topics the semi-structured interview method was selected due to its advantages. Semi-structured interviews combine structured and unstructured interviews to promote an exchange of opinions and ideas between both the interviewer and interviewee. This enables many topics to be explored through different perspectives, lending itself to a diverse range of interviewees and topics. This also enables opinions on topics, of which the interviewee may not be an expert on, to be explored.

Semi-structured interviews also lend themselves to a relaxing interview environment, stimulating interest in the research project and eliciting further information than traditional structured interviews [32]. The overhead, however, for performing these interviews is generally higher as the required research and planning before hand is required to be more in-depth.

The interview consisted of three primary topics. First, a discussion on security

in micromobility – its security concerns and aspects related to the unique security proposition of micromobility enterprise was explored. Second, EA modelling – its application and usage was explored. Finally, a discussion on the integration of security into EA.

As these interviews followed the semi-structured methodology these questions were subject to change during the interview based on respondents answers and queries, as such the interview outline provided in appendix A serves as a general framework, rather than an exact script used during the interviews.

3.2.3.3 Interview Analysis

Interview analysis was achieved through three steps. First, an automatic transcript, generated by the Zoom service, was retrieved. This transcript was then cleaned and edited for errors and readability. Finally, Excel was used to quantify each interview into relevant statements on pertinent topics. In total, three interviews were held with two interviewees from MaaS industry and one from a New Zealand government agencies transport engineering department. Each interview took an average of 43 minutes.

3.2.3.4 Limitations

Due to contextual difficulties – COVID-19 affecting availability, large enterprise mergers and general interview availability the sample size was negatively effected. This impacted the diversity of responses which makes it unlikely to provide a complete picture of industry. These effects were accounted for through strengthening the contribution of the SLR through the addition of the snowball mechanism, mitigating the lack of interview diversity, and increasing the SLR's contribution.

As the interview structure covered heterogeneous topics, the semi-structured interview format was selected providing the flexibility required when talking to domain experts. A limitation of this approach is the lack of comparable interview answers. This, however, was of little consequence when accounting for a small interview sample size.

As these enterprises are large, multi-national operations, their public facing customer service and contact portals were often non-responsive or not responsible for interview requests. Media contact portals offered similar service as interviews for academic purposes was not their responsibility. These factors increased the difficulty of contacting industry experts in these areas.

3.3 Information synthesis

Phase four provides a synthesis of information gained throughout phase two and three, answering the RQs and providing feedback on future research, trends, and opportunities.

The synthesis method consisted of utilising the qualitative analysis tool *Nvivo*. Nvivo enables the classification of textual information into *nodes*. Using this method, data extraction can be performed by following a predefined set of nodes and categorisations, enabling a collation of like data. Once collated, a descriptive synthesis on each node can be performed.

This method acts as the synthesis stage of the SLR, while also incorporating interview findings through the externalisation methodology. Further details on the synthesis method and outcomes are provided in Chapter 6.

Chapter 4

SMS

This chapter presents the SMS and is divided into three primary sections. First the SMS methodology is provided, identifying research questions, search strategy and data extraction processes. Next, the results are presented, classified by each research question. Finally a discussion and conclusion of the SMS is provided.

4.1 Introduction

As discussed in Chapter 3, a preliminary SMS study was held to educate the larger SLR study on its scope by identifying the current state of research. It also served as feedback on possible search strings, terminology, and database selection further minimising possible limitations of the SLR.

4.2 Methodology

An SMS was carried out in order to outline the current state of the art in the EA, ArchiMate, security and micromobility intersection. The guidelines proposed by Petersen et al. were followed [30], and a protocol was developed to support the execution of the SMS.

4.2.1 SMS Research Questions

The research questions (RQs) stated in this study and their motivation are as follows:

RQ1. How are security aspects being incorporated into ArchiMate EA models? To know what methods and techniques are being used to represent and enable security modelling in ArchiMate.

RQ2. What elements are required in ArchiMate to model security aspects in the context of micromobility? To identify a set of relevant security

concepts needed to be supported by ArchiMate in order to fulfil a security model of micromobility.

RQ3. What security pattern languages do micromobility companies currently employ? To identify the security pattern languages used in the industry.

RQ4. What support do architectural modelling languages provide for security in EA? To identify the reference architectures provided to support organisations when modelling their EA.

4.2.2 Identification of primary papers

The main terms of the phenomena under study were: EA, ArchiMate, Security and Micromobility. They were refined following an iterative approach based on pilot searches. To improve the quality of the search, common keywords were collated from directly relevant papers and standards such as ISO/IEC 25010 [33]. As suggested by Kitchenham [34], we approached experts in the field as well as used other means to increase the coverage of the keywords. The final terms used in the search string are shown in Table 4.1:

Table 4.1: Search string terms

Begin of Table	
Main terms	**Alternative terms**
EA	"Enterprise Architecture"
ArchiMate	TOGAF "architecture framework" "modelling language"
Modelling	"business modelling" "enterprise modelling" "conceptual modelling" meta-modelling ontology methodology

Continuation of Table 5.1	
Security	privacy dependability trust assurance integrity confidentiality accountability authenticity non-repudiation "risk management"
Micromobility	mobility micro-mobility "micro mobility" transport
End of Table	

The search scope was focused on peer-reviewed research papers published in journals, academic conferences, workshops and books.

The inclusion criteria followed for the selection procedure were:

- Papers that satisfy the search string.
- Journals, conferences and workshop papers.
- Papers written in English.
- Papers published up to January 2020 (inclusive)

The exclusion criteria considered were:

- Papers not focusing on security and EA.
- Papers available only in the form of abstracts or PowerPoint presentations.
- Papers that present a summary of a workshop presentation.
- Duplicated papers (same papers in different databases).

Scopus, IEEE Xplore and ACM Digital Library were used as the main search engines in order to preserve the quality of the papers. The fields used for retrieval were title, abstract and keywords.

The search was executed on January 15, 2020. The total number of returned documents was 987, with 883 from ACM Digital Library, 81 from Scopus and 23 from IEEE Xplore.

After applying the inclusion and exclusion criteria 14 papers were identified as primary papers. An expert on ArchiMate and security was consulted to recommend papers for manual inclusion, resulting in additional two papers that were considered for this review. However, none of them was accepted based on the inclusion and exclusion criteria.

4.2.3 Data extraction

For data extraction purposes a template containing fields for paper ID, authors, title, year of publication, publication title, document type, abstract, objective of the paper, security aspects mentioned, application domain, methods or techniques applied, and results was used.

The abstract of each study was analysed looking for keywords and their frequencies. Once the keywords were found, they were grouped in four categories: process area, application domain, security aspects and type of solution (see Table 4.2).

Table 4.2: Classification of keywords

Process area	Application domain
Planning / Management / Control Analysis / Design / Modelling Development / Implementation Validation / Evaluation / Assessment	CPS IoT
Security aspects	**Type of solution**
Trust Privacy Risk Interoperability Protection Access control Authorization	Methods Frameworks Languages Patterns ArchiMate EAM EISA

After performing the data extraction phase, a descriptive synthesis was provided in order to analyse each paper and discuss how they contributed to each research question or background information in relevant domains.

4.3 Results

The results are organised by RQ while the primary papers are categorised by their contribution to each RQs. Table 4.3 presents a mapping between primary papers and each RQs.

Table 4.3: Primary papers associated with the RQs

RQ	Primary papers
RQ1	[35] [36] [37]
RQ2	[38] [39] [40] [41]
RQ3	[42] [43]
RQ4	[44] [45] [46] [47]

Figure 4.1 shows the distribution of the papers per year and their publication type. It can be observed that in the last six years the concentration of publications has been at an average of 1.5 works per year. Only in 2016 no published studies have been found.

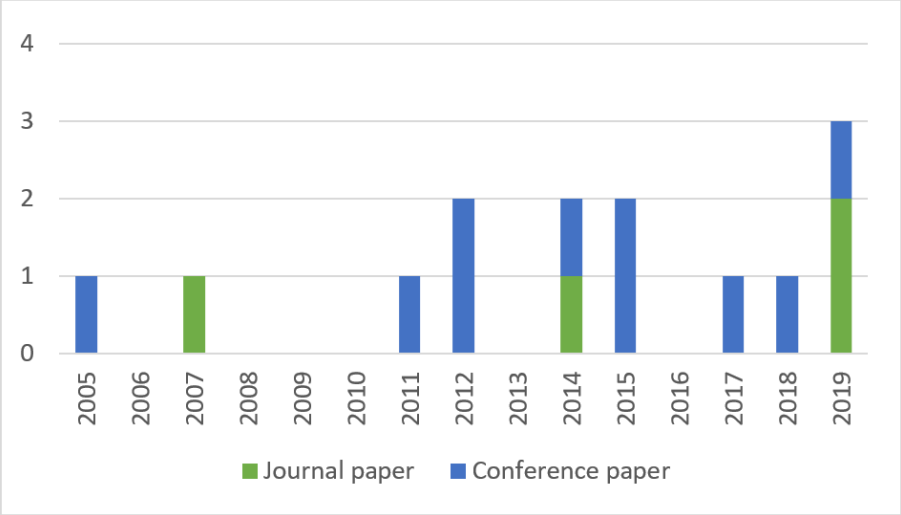


Figure 4.1: Distribution of studies per year and type

In the next subsections, the primary studies are analysed in regards to their contribution to each RQ.

4.3.1 RQ1. How are security aspects being incorporated into ArchiMate EA models?

Three primary papers provided insight on "How are security aspects being incorporated into ArchiMate EA models?".

The first two papers under analysis are authored by Korman et al., pursuing a general idea of leverage for ArchiMate to provide security. Korman et al. [35] investigate what data is required by a risk assessment (RA) process before it can be executed. RA is a common process in cybersecurity and provides an outline of different risk factors and a method to evaluate their influence on a system. Many studies provide analysis of RA methods from the perspective of features offered,

but none identified how much, or what kind of data was needed pre-execution [35]. Korman et al. provided a discussion on this topic and also explored how ArchiMate 2.0 could be used to model 12 RA process. The authors found that ArchiMate was able to represent 7 of the 12 methods investigated with nearly full coverage, however with varying confidence in their correctness.

Korman et al. [36] observed that authorisation and access control modelling was rarely supported by EA modelling languages. To this end, the authors developed a unified meta-model using ArchiMate which had the ability to describe a large set of access control schemes. The paper also offers a set of examples in which this meta-model is used to ascertain the correct configuration for access control.

In both papers, ArchiMate was used to model aspects of security RA and access control schemes, however, neither used the RSO framework. The earlier paper [35] was based upon the ArchiMate 2.0 architecture – before the RSO was released. The later paper [36] was released at a time when the RSO was available, however, it is not mentioned in the paper.

The third paper, by Mayer et al. [37], identified weaknesses in the Information System Security Risk Management (ISSRM) model. The ISSRM model is a method to identify and manage risks to the Confidentiality, Integrity and Availability (CIA) triad of an organisation's assets. Some examples of weaknesses were: verbose documentation, inability to describe complex enterprise and difficulty in maintaining continuous evolution. To improve upon these aspects Mayer looked towards EA management (EAM) to supplement ISSRM. The new model, EAM-ISSRM, was created by identifying a set of four EA modelling frameworks, such as ArchiMate, and performing a conceptual alignment of each to ISSRM. This comparison provided a set of concepts which were not included in ISSRM, and thus were likely to be responsible for the EAM languages superior performance in key areas. Drawing from this set, the authors identified aspects of EA modelling to incorporate and produced the EAM-ISSRM model. The model's performance was found to be superior to ISSRM, with better contextualisation, understanding of scope and maintainability of risk management processes.

Mayer et al. [37] started with a security model (ISSRM) and improved upon it by analysing EA modelling languages. This approach does not lend itself to identifying how security aspects can be incorporated into ArchiMate, however, a conceptual alignment is presented between ArchiMate 2.1 and ISSRM providing details on how ArchiMate and ISSRM compare in their operation. ISSRM is a security-focused domain model; therefore, this comparison provides insight on security concepts not included in ArchiMate, similar to the method used to identify concepts that were lacking in ISSRM.

These papers outline some of the ArchiMate's capability to represent security concepts; however, each paper identifies limitations in the security modelling implementation that TOG has proposed. It is also important to note that each paper utilises different versions of ArchiMate, ranging from ArchiMate 1.0 to 2.1. The current version of ArchiMate is 3.1, with flexibility in various areas. The improvements afforded by v3.1 will need to be identified when applying knowledge from these papers.

4.3.2 RQ2. What elements are required in ArchiMate to model security aspects in the context of micromobility?

Five primary papers provided insight on "What elements are required in ArchiMate to model security aspects in the context of micromobility?".

Anwar and Gill [38] explored how Digital Ecosystems (DE) – systems which support the interaction of people and third-party data service providers – can be modelled using EA languages. ArchiMate and six other modelling standards were compared to a well-established language in the DE domain – Adaptive Architecture Meta-Model (AAMM). This evaluation outlines how EA modelling standards often are unable to support DE modelling without modification due to DEs' wider scope. The authors define political and economic layers, which reside under the environment scope of DEs. This wider scope introduces complexity, which is not supported by many EA modelling standards. Anwar and Gill's paper outlines the differences between ArchiMate and other EA modelling standards such as SABSA, Agent-oriented modelling Language (FAML) and AAMM. A comparison between ArchiMate and SABSA – a well-known security architecture framework – can be drawn in order to evaluate potential elements and concepts that should be included in the ArchiMate language.

In a different paper, Pavleska et al. [39] developed a method to evaluate EAs based on the Electronic Simple European Networked Services (e-SENS), a reference architecture developed to provide cross-border and cross-sector digital services. The method's goal is to evaluate each architecture in terms of cybersecurity by analysing their performance regarding a set of relevant security goals provided by the Reference Model for Information Assurance and Security (RMIAS) and the European Union Agency for Cybersecurity (ENISA) guidelines. Pilot programs were run to test the validity of the method, evaluating different e-SENS architectures in regards to five aspects: security goals, countermeasures, information taxonomy, system security lifecycle and trust models. This paper demonstrates a method of evaluating

an architecture using a set of security guidelines - a method that could lend itself to identifying what elements ArchiMate needs to support for cybersecurity.

A desirable trait in computer systems and software is interoperability; however, interoperability often comes at the cost of security as they embody opposing quality attributes. In Shariati et al. [40], an analysis of a set of architectures and frameworks is provided, investigating how these frameworks unify interoperability and security. The selected frameworks all implemented versions of Enterprise Information Security Architecture (EISA) and were, by the authors' definition, holistic in their approach. Each framework was evaluated on three interoperability aspects: Technical, Organisational, and Semantic, with the authors concluding that security was often overlooked for interoperability performance. This paper identifies EISA as an important security reference architecture which should be investigated to ascertain security requirements, and therefore security elements for ArchiMate. The paper, however, does not describe in-depth the reasoning for evaluating frameworks as supporting or not supporting each interoperability aspect.

Solhaug and Seehusen [41] investigated risk management in critical infrastructure. Critical infrastructure offers specific challenges such as the need to continuously maintain the validity of the risk model. The authors offer a model which enables the continuous analysis and re-analysis of the target system by addressing only the risks that may be affected by any changes made. This streamlines the process and increases the speed at which risks can be evaluated. The model also facilitates on-the-fly modelling of syntactically correct CORAS diagrams [48] and supports arbitrary languages for target modelling, among other advantages. The CORAS risk modelling language is of particular note here as it defines a language (with extensions) that can present threat scenarios. A conceptual alignment of this language with ArchiMate would help indicate areas of improvement for ArchiMate and provide additional direction in the creation of security elements.

Alhadad et al. [49] implemented risk management by first defining a method for modelling a system – SOCIOPATH. The chosen modelling language is UML class diagrams where the authors abstract systems into two "worlds": first, the digital world, which is concerned with computer-centric processes, and second, the social world which is concerned with persons, physical resources and their relations. Alhadad et al. [49] then propose SOCIOTRUST, a method for producing a trust assessment of the model generated via SOCIOPATH, specifically by the derivation of formal rules that define relations between elements in the model. These rules are a complete way of defining the possible relations in the modelling language and are a suitable method for the development of any extensions to the ArchiMate language.

The five papers summarised above outline a set of methods that could be used

to elicit element requirements. A comparison or conceptual alignment could be done between ArchiMate and other modelling languages that provide support in the security domain, potentially identifying concepts that ArchiMate is lacking. From there new elements would be investigated to achieve the required representation. These steps should be taken, however, after the domain of micromobility has been investigated and its security requirements considered. As a whole, there seem to be many methods for improving frameworks and modelling languages, with most drawing upon other languages or frameworks that achieve the desired effect.

4.3.3 RQ3. What security patterns languages do micromobility companies currently employ?

Two primary papers provided insight on "What security patterns languages do micromobility companies currently employ?".

In order to evaluate what elements are required in a modelling language, an understanding of the domain to be modelled is paramount. This can be achieved through looking at pattern languages.

In the first paper, Hafiz et al. [42] attempted to collate security patterns found in software engineering into an ontology. In total, the authors unified 96 distinct security patterns as well as described the method they used to create the ontology. The authors discuss how patterns are harder to identify in the software domain as software is intangible and difficult to measure. Because of this, they call for other researchers to build upon their patterns to produce a more complete ontology. Using this ontology as an evaluation tool, an analysis of a modelling language and its coverage of the security domain can be conducted.

Pattern languages are more flexible as they do not utilise categories but rather connections between patterns – a type of ontology. The authors utilised pattern languages as other categorisation methods did not guide practitioners in selecting the correct pattern. Sources of information, modelling languages and the cataloguing of each pattern are all documented and would serve well as a reference for building a security pattern language around micromobility.

In the second paper, Janulevičius et al. [43] produced another security ontology, however, pertaining to a more specific cloud computing domain. Instead of evaluating patterns used by practitioners, the authors elicited requirements from sources such as the Cloud Security Alliance and other NIST recommendations. Many vendors in the cloud computing domain develop unique implementations of security that are not inter-operable because of their specific applications. This ontology seeks to standardise the vendors' approach in order to increase interoperability.

4.3.4 RQ4. What support do architectural modelling languages provide for security in EA

Four primary papers provided insight on "What support do architectural modelling languages provide for security in EA".

EA modelling languages are designed to present EAs in a digestible, informative and useful manner. Nakagawa et al. [44] presented the Knowledge-based Architecture Framework (KRAF), which provides a method to identify the knowledge found within reference architectures in the software domain. In this framework, knowledge is decomposed into four elements: domain, application, infrastructure and crosscutting elements, which are further decomposed into various attributes.

A reference architecture is considered complete when it provides coverage of all knowledge attributes. KRAF does not specifically address security in its components, rather, security is split among each of the elements. This supports the observation made by Demir et al. [45] in which security is often not explicit in architectural design, but is divided between components. As such, support for security elements in EA languages is often underdeveloped. The KRAF framework could be used to validate an EA language as the elements included in the language are required to be able to express the four elements of knowledge and their respective attributes.

Baloyi and Kotzé [46] introduced ICAMP (IoT and CPS Architecture-based Model for Data Privacy), a new reference architecture model designed to promote privacy on both the technological layer and the organisational layer. ICAMP is a promising reference architecture which enumerates upon many topics that are of interest. For example, its data management perspective helps ask and answer the questions mentioned above.

Demir et al., [45] and Pulkkinen et al. [47] consider transmission across boundaries at different levels of abstraction. Demir et al. [45] discussed the architecture of distributed systems and how security features are often not self-contained components but written manually into components of which security is a concern. The authors proposed a new architecture DISCOA, which implements a conceptual unit "aspects". Aspects can modify the architecture of a distributed system, delivering crosscutting functionality between components. This enables security to be removed from individual components and unified. In [47] the authors discuss communication at a higher layer of abstraction – between enterprises, and how security can be implemented at the interface between enterprises as secure communication between enterprises is beneficial to a large array of services.

4.4 Discussion

Although the SMS identified relevant background information on the state-of-the-art in ArchiMate, EA modelling and security, the results obtained outlined a gap in current academic research around security practices that micromobility enterprises employ.

The lack of research was expected as micromobility is a relatively young field of transport. Nevertheless, it has been observed that last year (2019) resulted to be the year with most number of papers published in regards to micromobility, which points to an increasing interest in the topic. This impetus of the research is being primarily generated by the transport sector, with more calls for papers regarding micromobility being issued [50]. This "bubble" of research coincides with a shift to alternative, sustainable transport initiating a mode shift in cities around the world [51].

The lack of research into micromobility and security, as found by this SMS, is due to the emerging nature of the topic in the security domain. Because of this, the results underwent the necessary adjustments to provide useful information by expanding the micromobility domain to its relevant components (i.e. IoT and information security). Since this SMS was performed, applicable papers have been published, illustrating the growing research on the topic. Vinayaga-Sureshkanth et al. published a paper discussing security considerations in the micromobility domain in 2020 [52]. The authors identify seven potential attack categories along with their suggested countermeasures. The limitation of the report however stems from its scoping of micromobility vehicle, solely focusing on e-scooters, and the fact – as identified by the authors – that micromobility is a quickly evolving domain.

Li et al. released a 'research-in-progress' paper in 2019, discussing privacy implications of micromobility enterprises [53]. The proposed study provides insight in the perception of privacy regarding e-scooter enterprise. This will help define the relationship between sharing economy businesses and privacy. Other recent research regarding micromobility focuses on exploitation examples of micromobility vehicles. Booth and Mayrany provide a report detailing the successful attacks against e-scooters [54]. The authors also provide a DREAD RA of each threat identifying spoofing and replay attacks as the most vulnerable.

Regarding RA and risk management, these resulted in the most common processes to be considered by EA modelling languages. The only provided support that ArchiMate features is that of the RSO. Even though security aspects must be considered when modelling an EA, the balance between simplicity and complexity is a

distinguishing characteristic of different modelling languages. Using the context of micromobility, ArchiMate's limited security modelling can be improved while also producing a timely security framework defining micromobility class vehicles and their associated security concerns.

In addition, security aspects related to IoT and CPS as well as current security practices performed by micromobility companies need to be investigated. Areas of interest include data security, access control, tamper resistance, information protection and RA, among others. There is discussion on using multiple modelling languages simultaneously when implementing EA to provide the holistic view of the company: BPMN for specific business processes and UML for complex low level processes [55]. This however requires an architect, or team of architects, to undergo significantly more training. The resulting model is also likely to be overly complex, making it difficult for the model to fulfill its purpose – a clear description of an enterprise that non-experts can analyze.

To mitigate validity threats, this SMS was based on sound procedures [30], [34]. A protocol for the study was created in order to minimize the impact of selection bias. The paper selection was carried out by the first author, with subsequent verification of the outcome by the second author. The same approach was followed for the data extraction process, where a template was created for extracting the verbatim data from each primary paper. Three databases were considered for mitigating the publication bias while a manual search was performed through a consultation with an expert in the field.

4.5 SMS conclusions and future work

The study revealed that the interest in modelling security in EA has increased in the last six years. However, the proposed modelling alternatives still present shortcomings related to the standardization, complexity and completeness.

There is a need for reference models, security standards and regulations in the context of micromobility in order to enable an accurate and effective representation through modelling languages. While the elements, extensions and frameworks found in the literature are mainly focused on dealing with RA, there is a need to design a methodology that allows enterprise architects to analyze and implement security aspects. Moreover, by providing a way for representing security aspects, further recycling and testing of existent solutions will be possible.

This SMS presented initial findings of the research area, providing valuable feedback on search strategy, research scope and expected results. Future work includes an in-depth SLR (Chapter 5) drawing from these findings in order to answer the

proposed RQs. There are four direct contributions this SMS provided for the future SLR. First, a general overview of the research contexts and current academic work in the area was provided. Second, analysis and procurement of the most representative search terms for the selected contexts. Third, the refinement of proposed research questions and, finally, improving search strings.

These contributions are utilised within the SLR to optimise its methodology, providing greater validity.

Chapter 5

SLR

This chapter presents the SLR methodology in detail, and its associated limitations. Next an analysis of the results is provided identifying trends of the retrieved primary papers and relevant statistics such as citation count. Finally, the body of this chapter provides the SLR's qualitative synthesis in which the primary papers are explored and discussed.

5.1 SLR Method

The SLR method carried out in this research followed the guidelines established in Kitchenham's report on SLR procedure [29] and was optimised with the previously run SMS chapter (Chapter 4) and the associated paper published at the Software Engineering and Advanced Applications (SEAA) conference in 2020 [1].

A primary influence the SMS scoping study had on the SLR was the widening of the micromobility domain to include parallel domains like IoT/CPS and connected vehicles. This decision was made after it was found that micromobility specific research was scarce in academia [1] opening another contribution opportunity for this research.

According to Kitchenham, the SLR method is composed of three phases. First; the *planning phase* in which the need of a review is identified and the review protocol is developed. Second; the *conducting phase* of the review executes the review protocol, selecting the primaries and providing data synthesis. Finally; the *report phase* of the review in which the conclusions and discussion on the data are provided. The *planning phase* will be discussed here, with the *conducting phase* described in section 5.3 and the *report phase* discussed in Chapter 6.

5.1.1 Planning Phase – Review Protocol

During the *planning phase* a review protocol was defined before the SLR execution, reducing the likelihood of selection, publication and other biases. Five components

are included within the review protocol:

1. Research Questions
2. Search Strategy
3. Study Selection Criteria
4. Data extraction strategy
5. Data synthesis strategy

5.1.1.1 Research Questions

The objective of the SLR was to answer the four primary RQs of the thesis.

RQ1: How are security aspects being incorporated into ArchiMate EA Models?

The SLR provides information regarding ArchiMate modelling techniques specifically in the domain of security modelling.

RQ2: What elements are required in ArchiMate to model security aspects in the context of micromobility?

The SLR provides information regarding relevant security concepts applicable to micromobility vehicles and services. This information will help inform what types of elements may be required to provide a security model of these systems.

RQ3: What security strategies do micromobility companies currently employ?

The SLR provides what mitigation processes and strategies that are likely to be used in the micromobility context.

RQ4: What support do architectural design languages provide for security in EA?

The SLR provides additional information on security modelling in the broader context of architectural languages.

5.1.1.2 Search Strategy

The search strategy identifies what the process of the SLR will look like and also what components of the process, such as the search string and data sources, will include.

Search Terms

The search terms were developed by identifying the primary topics covered in the RQs. Five topics were extracted:

- Enterprise Architecture
- ArchiMate
- Modelling
- Security
- IoT/CPS/Micromobility

From these topics, alternative and descriptive search terms were identified drawing from key words found in relevant literature, industry standards and finally, the previously mentioned scoping study.

Once the unique set of string terms were identified, analogous terms were identified for each term to decrease the possibility of relevant literature being excluded. For example, the term IoT has the analogous term "Internet of Things". Further analogous terms were identified through understanding each databases search algorithms, and their specific syntax rules. This lead to terms that involve hyphens having two additional analogous terms. An example of this is the term "non-repudiation" in Table 5.1.

Implicit analogous terms were also included through database inclusion mechanisms. For example, alternative spellings (US vs British spellings) are automatically included within the search query. For example, including the term "modelling" automatically includes its US counterpart, "modeling".

The final search string is shown below with Table 5.1 displaying the main search terms and their accompanying alternative terms.

("Enterprise Architecture" OR EA)
AND
(Archimate OR TOGAF OR "architecture framework" OR "modelling language" OR ontology)
AND
("business modelling" OR "enterprise modelling" OR "conceptual modelling" OR "meta-modelling" OR meta-modelling OR "meta modelling" OR modelling)
AND
(Security OR privacy OR dependability OR trust OR assurance OR integrity OR confidentiality OR accountability OR authenticity OR "non-repudiation" OR non-repudiation OR "non repudiation" OR "risk management")
AND
(IoT OR "internet-of-things" OR "internet of things" OR internet-of-things OR IIoT OR "industrial internet of things" OR "cyberphysical systems" OR CPS OR micromobility OR "micro-mobility" OR micro-mobility OR "micro mobility" OR transport))

Table 5.1: Search string terms

Begin of Table	
Main terms	**Alternative terms**
EA	"Enterprise Architecture"
ArchiMate	TOGAF "architecture framework" "modelling language"
Modelling	"business modelling" "enterprise modelling" "conceptual modelling" "meta-modelling" meta-modelling "meta modelling"
Security	privacy dependability trust assurance integrity confidentiality accountability authenticity "non-repudiation" non-repudiation "non repudiation" "risk management"

Continuation of Table 5.1	
micromobility	IoT “internet-of-things” “internet of things” internet-of-things IIoT “industrial internet of things” “cyberphysical systems” CPS “micro-mobility” micro-mobility “micro mobility” transport
End of Table	

Data Sources

Three databases were chosen for the initial search, mitigating publication bias and providing good quality material for the review. The chosen data sources were IEEE Xplore, Scopus and ACM DL.

Snowball Strategy

The SLR was extended via a snowballing mechanism to increase the scope of the review and to provide the most complete review possible. Forward snowballing retrieves all papers which reference the selected primaries of the previous iteration. In this way an exhaustive search is possible, identifying all referencing articles until no new articles are found.

In contrast to backwards snowballing, forward snowballing investigates progressively more recent publications – a benefit in regards to the chosen contexts as micromobility and EA are both recent phenomena.

Scopus was the chosen database for this method as it provides the tools necessary to identify referencing publications and has access to a large amount of research on appropriate subjects.

5.1.1.3 Study Selection Criteria

Selection criteria are divided between exclusion criteria and inclusion criteria. Three exclusion criteria and two inclusion criteria were defined for the SLR:

1. Exclusion Criteria
   (a) Research which are duplicate entries
   (b) Research that was categorised as book or book section results
   (c) Research that had no available full text
2. Inclusion Criteria
   (a) Research written and published in English
   (b) Research that presents information aligned with the SLR RQ's

Criterion 1a removes duplicate research results which were introduced either by multiple databases reporting the same research or the introduction of already excluded/included research by the forward snowballing process. Criterion 1b removes research categorised as a book or book section as the SLR is reviewing research papers (conference papers, journal articles). This provides research of reasonably comparable scope and contribution. Criterion 1c removes candidate research if the full text for the research was unavailable through university portals or open access.

Criterion 2a includes research presented in English due to time limitations and translation quality/accessibility. Criterion 2b includes research who's title, abstract and full text align with the RQs and goal of the SLR. When a study is able to pass all of the inclusion/exclusion criteria it is classified as a primary study for the review.

5.1.1.4 Data Extraction Strategy

Nvivo was selected as the primary data extraction tool due to its ability to classify qualitative data into contextual classifications. The data extraction strategy followed four steps. First, each primary paper was classified into four categories denoting which of the four RQs it contributed to.

Second, a thematic review of each RQ in regards to the primary papers was performed. Third, the discovered themes were codified in Nvivo, providing nodes of which sections of primary papers could be associated. An example classification scheme is shown in Figure 5.1).

Finally, an exhaustive review of each primary paper – its contributions to one or more themes and RQs were extracted for future synthesis. During this exhaustive review supplementary sub-nodes were provided ad-hoc enabling the most complete and valuable synthesis possible. Nvivo classification schemes for each RQ are provided in Appendix A.

5.1.1.5 Data Synthesis Strategy

Using the devised nodes for each RQ provided during the data extraction phase, synthesis is able to be achieved through performing a descriptive synthesis on each

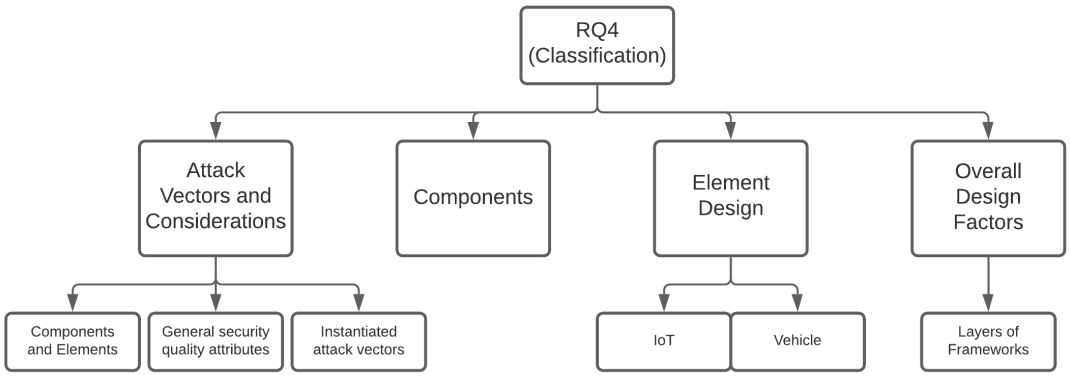


Figure 5.1: Classification scheme used to describe RQ2 themes

individual node. Each node contains the full set of information (as regarded by the researcher) on the associated topic and, during synthesis, can be broken into smaller individual nodes to aid in understanding and collation. As the returned primaries from the SMS scoping study (Chapter 4) were mostly heterogeneous and qualitative in nature, no strict meta-analysis process was pre-defined.

This process uncovers trends and information relevant to the RQ's, providing a platform for synthesis and discussion on the desired topics.

5.1.2 Limitations

Primary limitations of the SLR were addressed through the addition of the snowballing method, extending the search radius, providing an exhaustive mechanism and identifying state-of-the-art research. Other limitations potentially include lack of databases for the initial database search. The databases chosen for the SLR were three well regarded databases that the University of Canterbury had access too during the course of the research.

The selected snowballing mechanism's limitations include being limited to one database and not including backward snowballing. The effect of these limitations however is theoretically negligible as shown by [31], which identifies Scopus (the database used for snowballing) as being the database with consistently good recall and precision in terms of returned literature. Not including backward snowballing is negated by the relatively young field of research.

Non-English research was excluded through the inclusion/exclusion criteria, which may exclude relevant research. This was necessary due to the complexity associated with understanding non-translated works, and the time restraints associated with such a task.

As the scoping SMS identified much of the returned research as qualitative, quantitative analysis has been excluded from the SLR synthesis. Quantitative synthesis

in SLR's can be a useful way to measure metrics of a given domain and also provide a method of quality assessment. This leads to the difficulty associated with quality assessment in regards to heterogeneous, qualitative research. Quality assessment requires a like aspect or metric to be measured between selected primary papers which can then provide a relative assessment regarding the quality of the works [29]. This was implausible, as indicated by the SMS, as the heterogeneity of the search domains (Micromobility, Security, EA) made it very unlikely to identify like aspects between returned research. For this reason quality assessment was forgone.

5.2 Results and Analysis

Running the process as described above, a total of six snowball iterations were required after the initial search in order for no more primaries to be selected. The initial search yielded 943 references, with ACM DL providing 858, Scopus providing 56, and IEEE Xplore providing 28. A total of 27 were designated as primary papers and were used as the seed for the first snowball iteration.

The snowball results were as follows:

- Snowball one yielded 264 references from Scopus with 16 of these designated as primary papers.
- Snowball two yielded 175 references with 11 of these designated as primary.
- Snowball three yielded 72 references with 9 of these designated as primary.
- Snowball four yielded 26 references with five of these designated as primary.
- Snowball five yielded 19 references with three of these designated as primary.
- Snowball six yielded one paper which was not designated as primary, concluding the exhaustive snowball process.

A total of 72 primary papers were extracted for their contributions to answering RQs one through four. A detailed breakdown of this process and results can be found in Figure 5.7, providing metrics at each stage of analysis per snowball. Figure 5.7 also provides information on references that were excluded due to their full texts being unavailable through the University of Canterbury portals.

5.2.1 Meta analysis

The following meta-analysis consists of four sections. First, a discussion on the publication years of the primary papers and their trends. Second, predominant authors are outlined, providing an overview of authors who contributed the most to the set of primary papers. Third, primary paper types and their associated citation statistics are discussed, and finally, a contribution analysis is provided discussing the overall contributions of the primary papers to the RQs and topics.

5.2.1.1 Publication Year

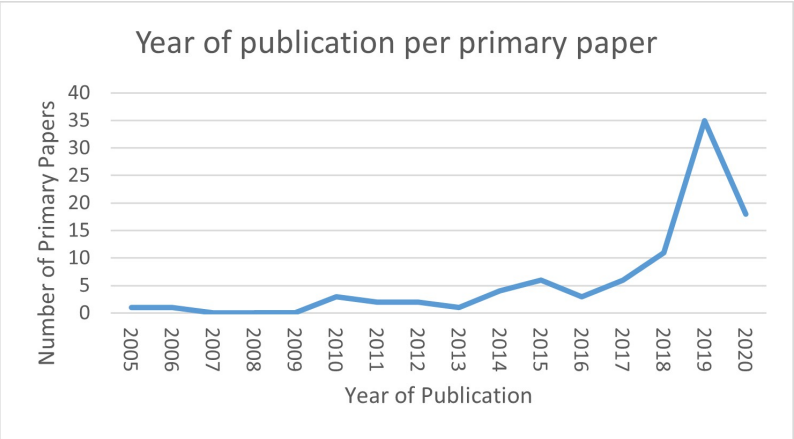


Figure 5.2: Publication years of primary papers

The overall publication years of the primary papers is provided in Figure 5.2. A general trend shows more recent papers being included in the study. There are two primary factors for this. First, forward snowballing was used during the collection process, which specifically identifies papers who cite the previous iterations primary papers. This method will always identify more recent papers than the previous primaries.

Second, the domain that is being investigated is relatively new. Information security and cybersecurity in general has gained traction over the last 10-20 years with security related to IoT and CPS applications being even more recent. EA, architectural modelling, and security modelling are also a more recent phenomena.

Supporting the duel nature of these factors, the initial search of the collection process (which does not rely on forward snowballing) identified papers from 2005 onward, highlighting the emerging nature of these domains.

Figure 5.3 identifies this trend in terms of RQ contribution. As expected *RQ1*, which relies on ArchiMate, only saw research contributions from 2014 onward, with an increase in the later years. This is due to the publication timeline of ArchiMate standards, and the release of the RSO, a TOG supported risk overlay for ArchiMate.

RQ2 and *RQ4* both have the earliest contributions in the form of some formative work regarding architectural modelling and information security. All RQs show an increasing amount of interest in the academic domain, as security and methods of governing security become more important in the industry domain.

5.2.1.2 Predominant Authors

Authors who were involved with the writing of 3 or more primary papers are presented in Figure 5.4. A more detailed table containing these authors and their associated publications can be found in Appendix B.

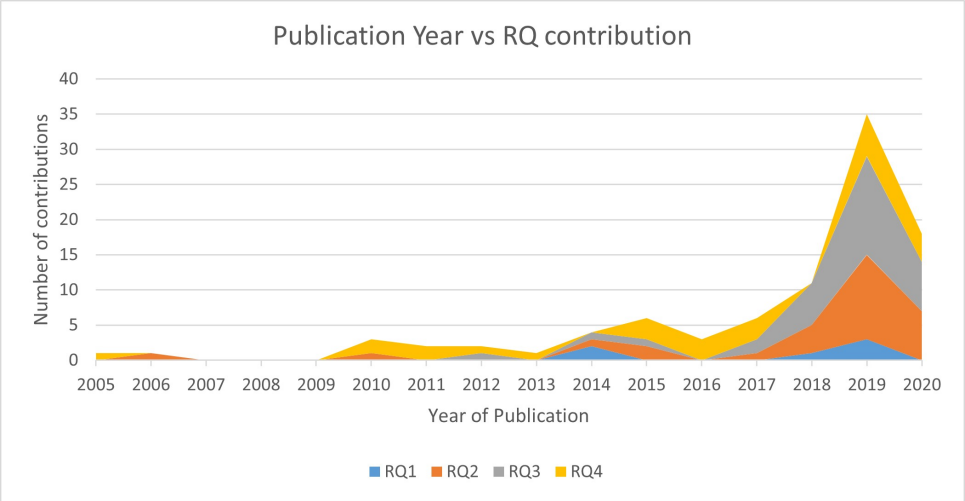


Figure 5.3: Contributions per year per RQ

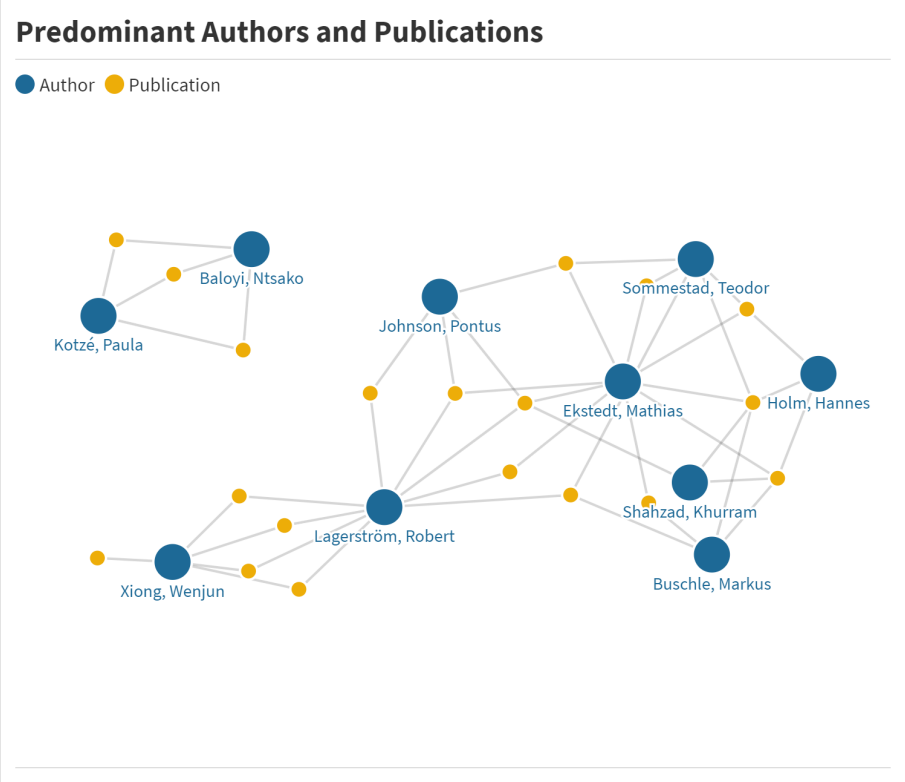


Figure 5.4: Predominant Authors and their associated publications

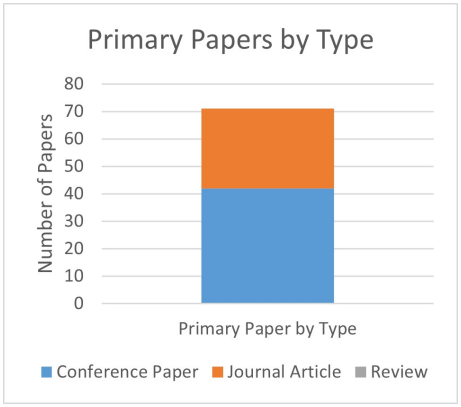
Figure 5.4 visualises all publications that each author worked on. As can be seen, these authors who contributed three or more primaries often worked on joint papers, publishing 19 of the 71 primary papers in this study. Two of these authors, Lagerström and Mathias, where involved in nine papers each, co-authoring four papers together on the topic of security modelling, and security modelling tools.

5.2.1.3 Primary Paper Types and Citation Statistics

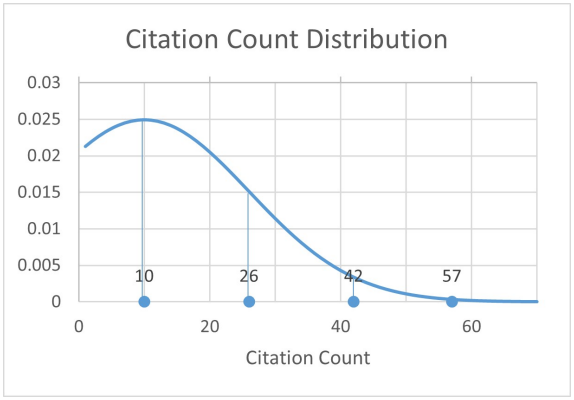
A total of 29 papers were journal articles and another 42 were conference papers (Figure 5.5a).

The citation statistics for this section were collected from Scopus on February 04, 2021. The average citations for the primary papers was 9.9 citations per paper, with a median of four. A total of 12 papers had zero citations while seven papers had over 20 citations each. The citation distribution is provided in Figure 5.5b and shows the majority of primary papers maintain ten citations.

Papers with zero citations tended to be recently published, with nine of the 12 papers being published in 2019 and 2020. This may explain the lack of citations of these works.



(a) Categorisation of Primary Papers



(b) Distribution of citations over all primary papers

Figure 5.5: Primary paper type and citation statistics

5.2.1.4 Contribution Analysis

This section describes major contributors to the SLR, as well as a general overview of the contributions in each RQ.

5.2.1.4.1 Overall Contributions

In total, six primary papers contributed to *RQ1*, 29 to *RQ2*, 32 to *RQ3* and 26 to *RQ4*. A set of **22** primary papers contributed to more than one RQ - a more detailed analysis is provided in Table 5.3. Figure 5.6 provides an overview of the percent contribution by primary paper to each RQ.

RQ1 has relatively few primary papers associated with it due to its narrow scope. *RQ1* specifically identifies papers discussing ArchiMate in the context of security modelling, a topic with a more direct scope than the other RQs.

RQ2 and *RQ3* were found to account for a substantial portion of the 22 primary papers who contributed to more than one RQ. This is due to the similarity of their

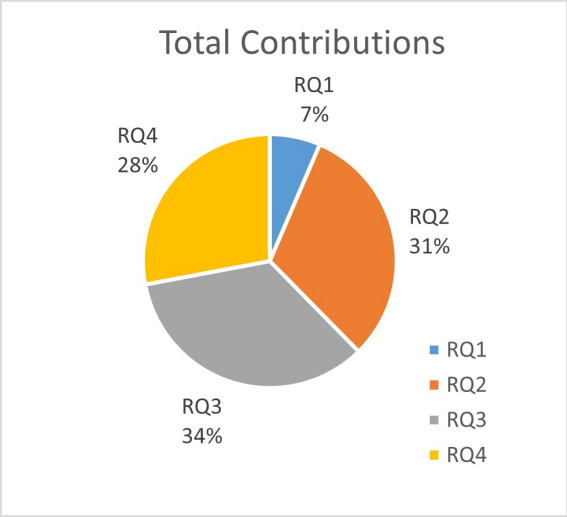


Figure 5.6: Contribution percent per RQ

domain - many papers which identify security requirements of systems (and thus help answer *RQ2*) also provide overall frameworks and security methodologies (helping answer *RQ3*).

Finally, *RQ4* exhibits the most self-contained papers, that is the primary papers referenced within this RQ are not shared across RQs, and only appear once within sub-sections of the RQ (Table 5.3). The scope of *RQ4* contained that of *RQ1*, identifying generally what security mechanisms architectural description languages afforded, and thus more primary papers were associated with it.

5.2.1.4.2 Topics by contribution

Metrics can be derived from each topic by analysing the number of primary papers which contribute to each. Through this method, topics with the highest contributions and lowest contributions can be identified.

Table 5.2a identifies sections and topics which received the most contributions from the primary papers. These sections constitute the Method of Security (3.2.1, e.g. RQ3, Topic 2, Section 1), Attack Vector Considerations (2.1.3), Element design (2.2), and Security Standards (3.3.1). In general, RQ2 and RQ3 contain the most contributions in this research as discussed above.

Sections which contained the least contributions were Application domains (2.1.5), Architectural Standards (3.3.4), Other Standards (3.3.5), and Literature Reviews (4.5.1), see Table 5.2.

These findings, while indicating sections with high and low levels of contribution do not indicate a definite cause for this, as categorisation bias can play a major role in the classification of contributions. As such, this section only intends to identify sections of dominance within this research, and perhaps indicate the possibility of areas of research which are more common than others but does not conclude this.

Table 5.2: Highest and lowest contribution by topic

(a) Highest Contributions

RQ section	Total Contributions
3.2.1	17
2.1.3	10
2.2	9
3.3.1	9

(b) Lowest Contributions

RQ section	Total Contributions
2.1.5	2
3.3.4	2
3.3.5	2
4.5.1	2

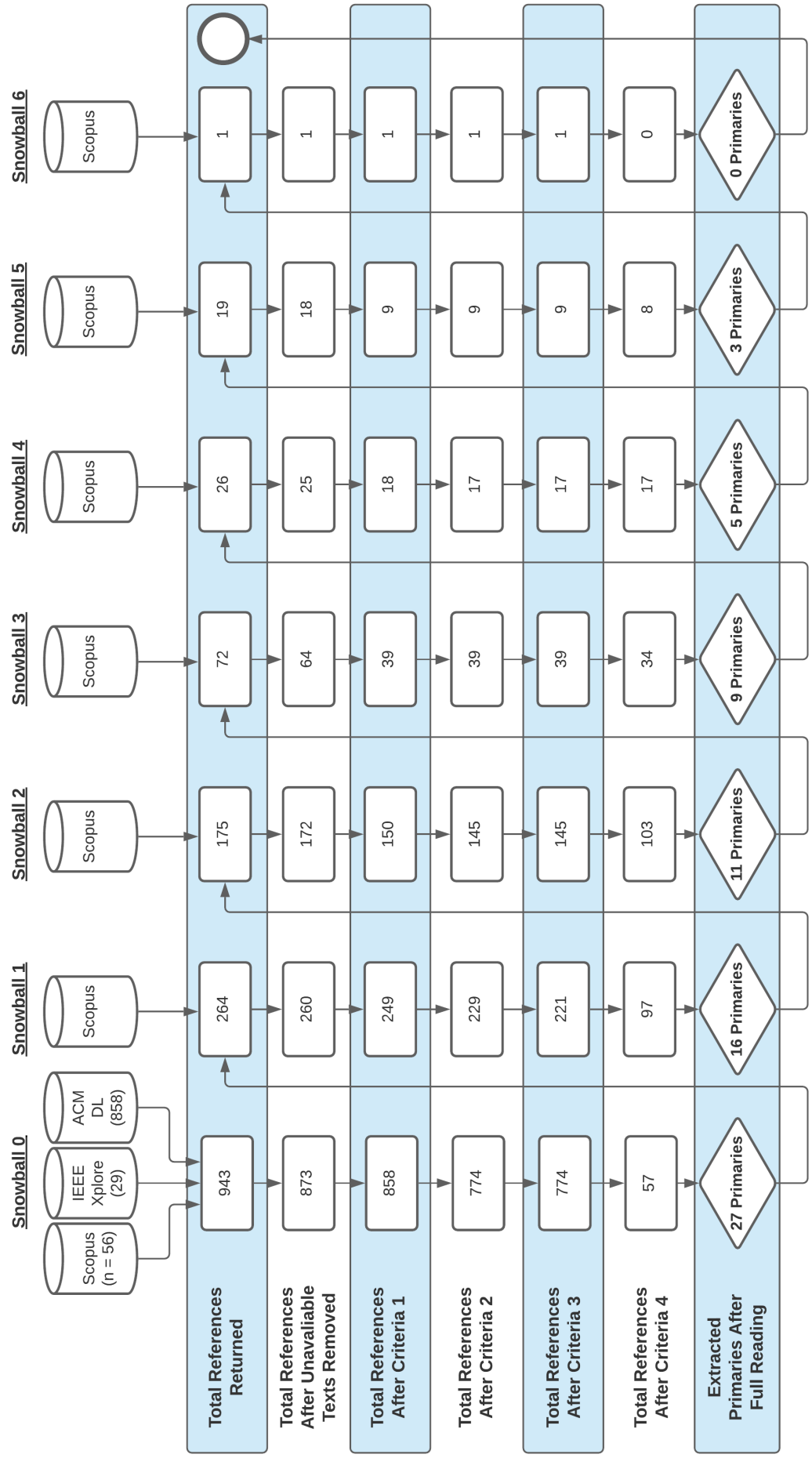
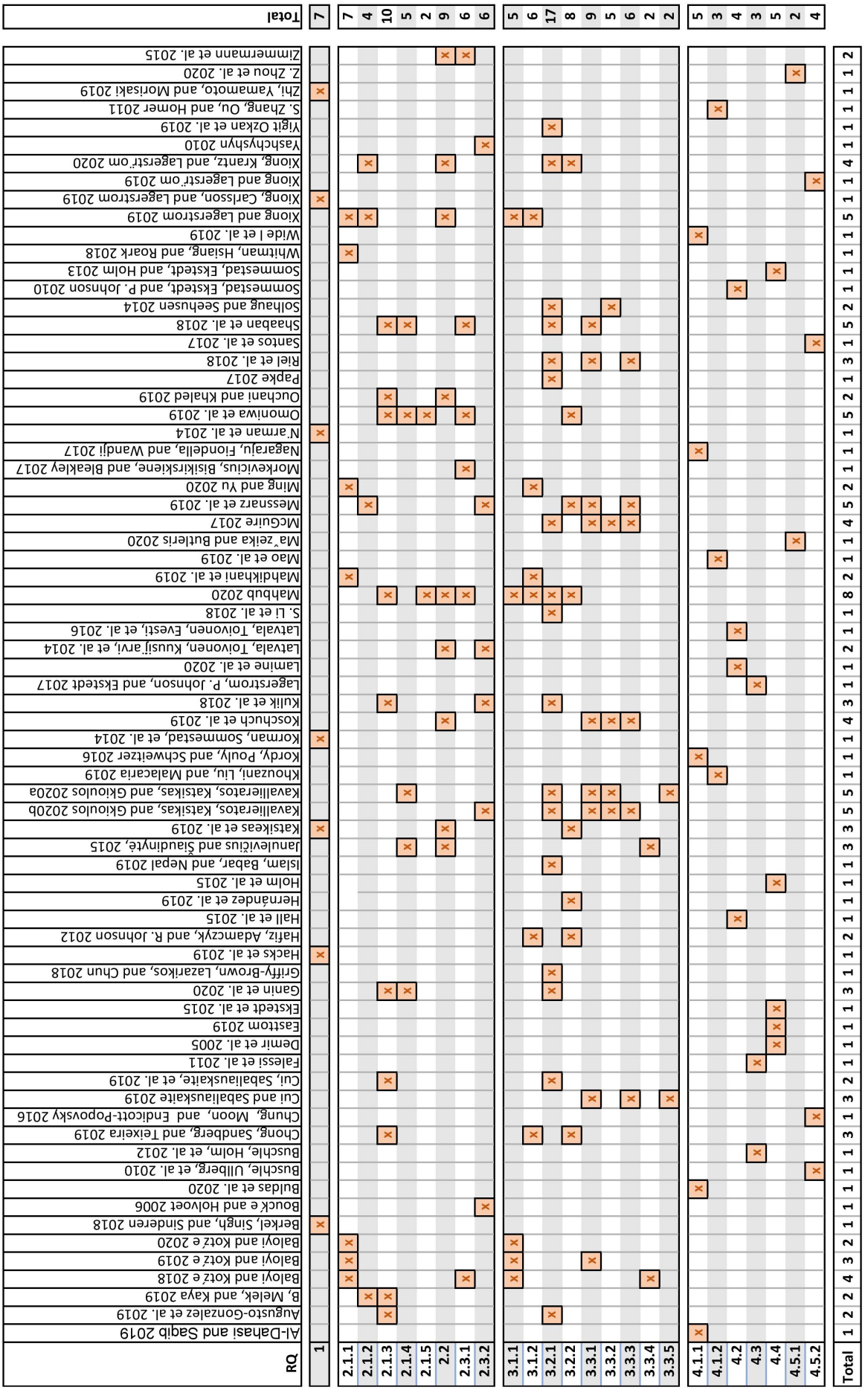


Figure 5.7: SLR process including snowball extension

Table 5.3: Contribution of each primary paper to each RQ



5.3 Qualitative Synthesis

The qualitative synthesis is structured by RQ, describing primary papers and their contribution. The RQs are grouped by like themes with RQ2 and RQ3 being presented together, discussing security and micromobility, and, RQ1 and RQ4 discussing Archimate and modelling.

Each RQ will consist of smaller, contextual contributions which the primary papers provide. Each of these contributions will have contextual analysis provided to emphasise their contributions to the RQs.

5.3.1 RQ2: What elements are required in ArchiMate to model security aspects in the context of micromobility?

To provide distinct contributions to RQ2, three topics were identified of which various primary papers provided information on. Each topic is further enumerated by its components, which were developed to provide meaningful contributions to the RQ.

These topics and their components are as follows:

1. Attack Vectors and Considerations
    - Privacy
    - Vulnerable Components
    - Attack Vectors
    - Security Quality Attributes
    - Application Domains
2. Element Design
3. Overall Design Factors
    - Layers of Frameworks
    - Design Objectives and Considerations

These topics, and their components, are educated by primary paper contributions, and contain an aggregated contribution and general discussion of each contributing primary.

5.3.1.1 Topic 1: Attack Vectors and Considerations

The **19** primary papers which contribute to this topic are best described with five components:

1. Privacy
    - Discusses attack vectors and considerations in regards to privacy
2. Vulnerable Components
    - Discusses vulnerable components provided by literature
3. Attack Vectors
    - Discusses common attack vectors discussed in literature
4. Security Quality Attributes
    - Discusses quality attributes identified in literature
5. Application Domains
    - Discusses the application domains of IoT/CPS technology, and its vulnerable application domains

Privacy, as a contribution, was divided from the general contribution "Attack Vectors" due to its prominence in the returned literature. Privacy is also distinct from traditional vulnerabilities as it is more an implicit goal than an instantiated vulnerability - separating these contributions provides a clear distinction between them and also offers more detailed analysis of both.

Vulnerable Components were extracted as a contribution as attack vectors are often directly related to certain technologies which are included in certain technological components. For example, the concept of an Electronic Control Unit (ECU) which are commonly used in smart vehicles often provide the vulnerabilities which bad actors exploit. By extracting the components, a more detailed understanding of security and vulnerability can be created.

Attack Vectors identifies all non-privacy attacks on IoT/CPS systems and their associated application domains. This provides real world attacks and scenarios, identifying possible vulnerabilities in the micromobility application domain.

Security Quality Attributes identify the goals that security elements and components are attempting to achieve. These help define the contextual goals of security in different application domains.

Application Domains extracts information provided by the primary papers regarding security critical application domains of IoT/CPS technologies.

5.3.1.1.1 Privacy

Five primary papers discuss privacy in the context of attack vectors and considerations.

Ntsako Baloyi and Paula Kotzé contribute heavily to this subtopic with three of their papers being included. Their least recent paper, published in 2018, discussed privacy considerations in relation to IoT and CPS technologies [46]. IoT and CPS technologies, by nature, can be problematic in regards to the possible covert nature of data collection. Because of this, a subject may have not given consent to the collection of their data. The authors cited aspect of IoT and CPS privacy concerns is the natural ability for these technologies to capture and/or record vast amounts of personal information at a scale that has not been implemented before. This vast amount of data can violate personal privacy in ways that were not previously possible. One method that can violate personal privacy through mass data extraction is the method of inference - a simple example of this would be "Shadow Profiles" on Facebook which can identify an individual without them ever creating an account [56].

The duos next paper, published in 2019, provides a set of data privacy guidelines for organisations in the application domain of CPS and IoT [57]. The authors discuss an intrinsic right for an individuals privacy, and the trust individuals place in organisations to respect this right. CPS and IoT data lifecycles are explored as the authors stipulate that data privacy necessitates the investigation and design of data lifecycles in order to address privacy concerns. There is a large variance in how data lifecycle is handled between technological domains. CPS specific lifecycles are comprised of three phases - data preparation and persistence, data visualisation and core data analytics, and data exchange. This varies from an IoT lifecycle which can be categorised into eight stages - create, collate, cleanse, store, archive, retain, process and purge. These lifecycle frameworks offer a perspective on potential privacy risk in the lifecycle process, which can further be used to identify security considerations.

Finally, Baloyi and Kotzé's most recent paper, published in 2020 offers a report on data privacy, and what benefits are afforded to organisations that take a preemptive approach to complying with regulatory policy [58]. While the work is focused on the reasoning behind privacy compliance, the authors offer context regarding the importance of privacy in the CPS and IoT application domains. These domains intrinsically collect an process vast amounts of personal information. This is done to provide utility in various applications however it increases the risk to privacy breaches. The authors identify that privacy compliance is often seen as an extra burden to organisations, resulting in negative consequences and lost opportunities.

Generally, these three papers identify the overall characteristics of the IoT and CPS domain which cause an elevated risk to privacy, namely - the opportunity for the incognito recording of peoples information and the opportunity to record vast amounts of information. The pairs later work identifies how and why regulatory policy should be enforced within an organisation and encourages enterprises to do

so.

A more technological perspective regarding how emerging technologies are affecting privacy risk is offered by Whitman, Hsiang and Roark [59]. They described how the mixture of machine learning, big data and ubiquitous data collection have lead to discriminatory outcomes with disparate impact and material consequences. The authors offer the concept of collaborative development between enterprises, the public and governing bodies in which transparency enables healthier data practice in industry.

Finally, [60] discussed the threat modelling of connected vehicles, and how these can be utilised to provide privacy analysis. During this work the authors defined four types of vehicle data in order to identify how privacy can be compromised. These four types of data are geolocation data, vehicular sensor data, biometric data and behavioural data. A breach of geolocation data would provide an adversary with the location of the vehicle, vehicular sensor data is susceptible to false data injection attacks, biometric data contains personally identifiable information used for security purposes and behavioural data can be misused by an adversary to infer private information about an individuals life.

5.3.1.1.2 Vulnerable Components

Four primary papers provide discussion on non-generalised vulnerable components. All papers discussed in this section were written for the vehicle application domain. The components provided by the primary papers are presented in List 5.3.1.1.2 below.

List of non-generalised vulnerable vehicle components:

- Airbag System [61]
- Keyless entry [61]
- TCU [61]
- Bluetooth [61]
- OBD-II [61]
- ECU [61] [62]
- Steering system [63]
- e-powertrain system [63]
- Break system [63]
- Geolocation Data [60]
- Vehicular Sensor Data [60]
- Biometric Data [60]
- Behavioural Data [60]
- Software [60]
- Protocol [62]
- Wireless component [62]

Melek and Kaya [61] provides a list of security vulnerabilities found in vehicles and their affected components. Some examples of vulnerable components are the Airbag System, Keyless Entry, Telematics Control Unit (TCU), Bluetooth, OBD-II and ECU Gateways. A unique aspect of vehicle attack vectors as discussed by the

author is the inclusion of an OBD-II port, required to be installed in all cars in the U.S. from 1996. The authors found that this port had the largest number of vulnerabilities out of all considered components making this a unique security risk to cars.

Messnarz et al. [63] considers fail operational states in highly AVs. Specifically, the authors discuss three AV systems - Steering, E-Powertrain, and the Break System. These systems are discussed in the context of fail-operational states - that is, systems that continue to operate when their control systems fail.

Xiong et al. [62] explores threat modelling and attack simulations in the application domain of connected vehicles. The authors provide the primary assets that a connected vehicle consists of - ECU, SoftwareProduct, Dataflow, Protocol and Network. These critical components are vulnerable to cyberattacks as shown by ethical hacking performed on the 2014 Jeep Cherokee and 2015 Cadillac Escalade vehicles, enabling in one case, full remote control over critical systems.

Xiong and Lagerström [60] discuss threat modelling. they provided four types of vehicle data in order to identify how privacy can be compromised. These four types of data are geolocation data, vehicular sensor data, biometric data and behavioural data. Aside from these data classifications the authors also provide a definition of the main technological assets of connected vehicles. The four primary assets are provided - ECU, Network, Data and Software.

5.3.1.1.3 Attack Vectors

Ten primary papers discuss attack vectors. The attack vectors and considerations provided by these papers are summarised by Tables 5.4, 5.5 and 5.6. These provide each attack vector from the perspective of what domain it is initiated from (social, physical or cyber) and also provides what application domain the attacks are discussed in.

Two of these primary papers identify attack vectors and considerations within the CPS application domain.

Chong et al. [64] offers a tutorial introduction to security and privacy in the CPS application domain, providing an overview of security concerns and possible attack vectors. The authors describe four key attack scenarios in the CPS context - eavesdropping, open-loop false-data injection, replay and denial-of-service (DoS) attacks. Eavesdropping attacks have an adversary secretly collecting information from a network and therefore violating the confidentiality of this data. Open-loop false-data attacks involve the adversary corrupting the integrity of the transmitted data. Replay attacks have the adversary attempt to achieve a desired effect by recording

Table 5.4: Attack Vectors identified in the social domain

Domain	**Attack Vector**
IoT	Privileged Insider Attack [65]
RA	Personnel Background [66]
	Awareness and Training [66]
	Access Control [66]
	Loyalty and Well Being [66]

Table 5.5: Attack Vectors identified in the physical domain

Domain	**Attack Vector**
AV	Direct Physical Attacks on internal sensors [67]
	Direct physical attacks on ECUs [67]
	Direct attacks on Controller Area Network (CAN) bus [67]
	Direct physical attacks on on-board computer [67]
	Direct physical attacks on external sensors [67]
	Direct physical attacks on Human Machine Interface [67]
	Direct physical attacks on brought-in devices [67]
	Attacks via USB [67]
DBaaS	Physical access control to hardware [68]
IoT	Damage and modification of hardware [69]
RA	Physical access control to hardware [66]
	Obsolete Hardware [66]
	Counterfeit Hardware [66]
	Portable Devices (USB) [66]
CPS	Resource Depletion (CAPEC-119) [70]

Table 5.6: Attack Vectors identified in the cyber domain

Begin of Table	
Domain	**Attack Vector**
RA	Logical Access [66]
	Obsolete Software [66]
	Counterfeit Software [66]
	Antivirus and Scan Coverage [66]
DBaaS	DoS [68]
	Access control [68]
	privacy leakage [68]
	Co-located application data [68]
	Modification of configuration [68]
SV	Protection mechanism failure [61]
	Buffer exploits [61]
	Information leaks/disclosure [61]
	Cryptographic issues [61]
	Permissions, privileges, and access control [61]
	Input validation [61]
AV	Deception and DoS attacks on internal sensor measurements and control actuations [67]
	Deception and DoS attacks on inter-ECU communications [67]
	Deception and DoS attacks on CAN bus communication with ECUs [67]
	Deception and DoS attacks on CAN bus communication with on-board computer [67]
	Attacks on Ethernet [67]
	Attacks on WiFi [67]

Continuation of Table 5.6	
CPS	DoS attack [64] Replay attack [64] Eavesdropping [64] Spoofing (CAPEC-156) [70] Data Leakage (CAPEC-118) [70] Open-loop false-data injection attack (CAPEC-152) [70] [64] Exploitation of Authentication (CAPEC-225) [70] Fuzzing (CAPEC-28) [70]

Continuation of Table 5.6

IoT	Node Capturing [65]
	Malicious Code Injection [65]
	False Data Injection [65]
	Side-Channel Attacks [65]
	Eavesdropping and Interference [65] [71]
	Sleep Deprivation [65]
	Booting Attacks [65]
	Phishing Site Attack [65]
	Access Attack [65]
	DDoS/DoS Attack [65] [69]
	Data Transit Attacks [65]
	Routing Attacks [65]
	MITM Attack [65] [69]
	SQL Injection Attack [65]
	Signature Wrapping Attack [65]
	Cloud Malware Injection [65]
	Flooding Attack in Cloud [65]
	Replay Attack [65]
	Impersonation Attack [65] [71]
	Data Thefts [65]
	Access Control Attacks [65]
	Service Interruption Attacks [65]
	Malicious Code Injection Attacks [65]
	Sniffing Attacks [65]
	Reprogram Attacks [65]
	Attacks on V2X network [65]

End of Table

a previous interaction and executing this interaction at a later date. Finally, DoS attacks seek to tie up all available resources of a system by sending many superfluous requests of which the system must service. The authors also provide a quick discussion on undetectable and stealth attacks in which specific types of false-data attacks are used. These attacks are designed to be analogous with the natural behaviour

of the system, and in that way are stealthy as the system may have no means of identifying if it is under attack or not.

Ouchani and Khaled [70] discuss the modelling of threats in the CPS application domain. When generating their threat behaviours for the CPS system the authors identify six primary attack vectors: Spoofing, Data Leakage, Resource Depletion, Injection, Exploitation of Authentication, and Fuzzing. These attack behaviours were extracted from the Common Attack Patterns Enumeration and Classification assurance strategic initiative (CAPEC) and are classified as software attacks and communication attacks.

Five of these primary papers identify attack vectors and considerations within the IoT application domain.

Mahbub [65] provides a considerable study on IoT security, identifying many vulnerabilities and considerations. A generalised IoT framework is provided which describes IoT infrastructure in five layers. Using this framework the authors classify attack vectors by layer, identifying seven vectors in the sensing layer, five in the networking layer, eight in the middleware layer and six in the application layer. The authors provide mitigation techniques in the gateway layer however do not provide attack vectors or considerations for this layer. Example attack vectors are Man-In-The-Middle (MITM) attacks, network flooding attacks, and code injection attacks.

Kulik et al. [71] considers a threat-driven security framework for IoT Systems, specifically designed to provide verification on the systems security posture. During the design of this framework the authors identified two probable attack vectors of IoT systems. The first attack vector is eavesdropping - the act of an adversary secretly obtaining information and data from a network. The second attack is identity faking - when an adversary poses as a trusted source or user in order to trick the system into servicing their phoney requests.

Shaaban et al. [68] discusses the proposed IoT reference model CloudWoT for the IoT application domain. The authors discuss security challenges and attack vectors in relation to the model, drawing on the traditional CIA triad. Further attack vectors are identified, specifically in the sub-domain, Database as a Service (DBaaS) context. DoS, physical attacks, modification of configuration, access control, privacy leakage, and co-located application data are mentioned as being vulnerable attack vectors of DBaaS. The authors briefly mention vulnerabilities associated with the semantic web, stating that several attacks in this context can disclose sensitive information - violating privacy.

Augusto-Gonzalez et al. [72] presents the "GHOST" security framework. The authors cite the heterogeneous, dynamic and internet connected nature of IoT environments as some of the leading aspects that make smart homes vulnerable. During the validation phase of the new security framework "GHOST" the authors propose three attack vectors. First, physical attacks in which physical damage is caused by the removal of the battery, shut down of the device, breaking of the device, injection of another device into the network, or mechanical exhaustion of physical buttons leading to the inoperability of the device. Second are network attacks in which network traffic is tampered with in order to provide a malicious affect. This includes impersonation attacks, sniffing attacks, DoS attacks, as well as many other well known network exploits. The last category is software attacks in which an adversary gains access to a device within the network and either modifies or installs new software to achieve a desired effect. These attacks can often enable network attacks such as compromising a gateway to achieve a form of eavesdropping.

Omoniwa et al. [69] explores the merits of the Fog and Edge Computing (FEC) based IoT (FECIoT) architecture in regards to its security issues. The authors identify three attack vectors in FOCIoT the architecture - Distributed Denial of Service (DDoS), MITM, and physical attacks. DDoS is specifically effective against FECIoT as the availability of these systems is usually critical to their application. This attack is also relatively easy to implement once a sufficient bot net has been acquired. MITM attacks are particularly damaging to privacy as they have the ability to disclose sensitive information, such as the location and identify of FEC devices. The FECIoT application domain is particularly susceptible to this as, due to the devices resource constrained nature it has have particularly weak encryption for communication. Finally, the physical attack vector consists of a bad actor(s) damaging/modifying the devices in order to achieve their goals.

Two of the primary papers identify attack vectors and considerations in the vehicle application domain.

Cui et al., [67] explores security considerations in the application domain of AVs. The authors provide two overarching categories of attacks in this domain - physical and cyber. Further they identify two primary cyber-attacks - deception attacks and DoS attacks. Deception attacks are when an attack is able to use unauthenticated data to deceive vehicle systems or other entities. DoS attacks have particularly high risk in the context of AVs. If jamming is successful, real-time information will be delayed which can impact the behaviour of the vehicle. The authors identify vulnerable components in the AV ecosystem and 16 distinct attacks that can affect these systems. These range from direct physical attacks and modifications of internal sensors/actuators, to deception and DoS attacks and are included in tables 5.4, 5.5 and 5.6.

Melek and Kaya [61] provide a list of security vulnerabilities found in vehicles and their effected components. The authors analysis provided information on the top 25% of Common Weakness Enumeration (CWE) consisting of protection mechanism failure, buffer errors, Information leak/disclosure and other attack vectors.

Finally, **one** of the primary papers identified attack vectors and considerations in the context of RA.

The last primary paper in this section discusses a proposed decision framework for cybersecurity RA which quantifies threats, vulnerabilities, and consequences [66]. The authors define threats as "a person or an organisation that intends to cause harm". This is further quantified into two factors - ease of attack and benefits of a successful attack. These factors were described further below:

- Ease of attack - the perception of how easy it is to carry out an attack. This perception is realised through three factors:
  - Information held by the attack regarding the target system.
  - Technology that the attacker has access too.
  - Delivery options available to the attacker.
- Benefits of a successful attack - net gain for an attacker. This gain is realised through three factors:
  - Financial gain
  - Political gain
  - Other gains

The authors also enumerate upon the possible vulnerabilities within the cybersecurity landscape. Three overarching vulnerability domains are classified; The physical domain, information domain and social domain.

5.3.1.1.4 Security quality attributes

Five primary papers provide discussion on general security quality attributes.

All five primary papers include the traditional security attributes - CIA, however some more specific attributes were also offered (Table 5.7).

Janulevičius and Šiaudinytė [73] considers security diagnostics in distributed systems. The authors characterise security challenges in this area with eight aspects - identification, entity authentication, data authentication, authorisation, integrity, confidentiality, non-repudiation and execution safety. The authors then design a

Table 5.7: Security attributes extracted from primaries

Unique Quality Attributes	
Domain	**Security Attribute**
DS	Identification [73]
	Authentication [73]
	Authorisation [73]
	Non-repudiation [73]
	Execution safety [73]
IoT	Trust [69]
	Authentication [69]
	Privacy [69]
	Access Control [69]
CPS	Authenticity [74]
	Possession and Control [74]
	Utility [74]
	Non-Repudiation [74]

Like Quality Attributes		
Domain	**Security Attribute**	**Articles**
DS	Confidentiality	[73]
IoT	Integrity	[69]
CPS	Availability	[74]
RA		[66]
IoT		[68]

security device which consists of four primary elements - Security Agent, XML-Binder Agent, Admission Agent and Queue Agent.

Omoniwa et al. [69] explores the merits of the FECIoT architecture in regards to its security issues. The authors identify seven considerations; trust, authentication, integrity, confidentiality, privacy, availability and access control.

Trust, in an FECIoT application, is more of a state that an element or node can achieve than a quality it possesses. Elements require sufficient security mechanisms promoting them to trusted elements within an IoT network.

Next, authentication in an FECIoT application involves unique constraints. IoT devices are often resource constrained, making the resource heavy private-public key method cumbersome. This requires specific authentication methods and protocols

to be implemented in this context. Availability is also of unique significance in this application domain due to the correlation between this domain and latency-sensitive applications. Integrity, confidentiality, privacy and access control considerations are all similar to other technology.

Ganin et al. [66] propose a decision framework for cybersecurity RA which quantifies threats, vulnerabilities and consequences. To measure the impact or consequences of a successful attack the authors use the CIA triad. Vulnerability's and attacks provided by the authors were then measured against these metrics.

Kavallieratos et al. [74] provide a method which elicits both security and safety requirements in the CPS application domain. To do this the authors define seven security attributes and 11 safety attributes. These attributes are used in the process of elicitation, providing aspects of the subject that need to be evaluated. The seven security attributes and 11 safety attributes were defined as shown in Table 5.8 below.

Table 5.8: Security and Safety objectives defined by [74]

Begin of Table	
Security Attributes	
Attribute	**Definition**
Confidentiality	Information exchanged, and communication links between CPSs and services offered by CPSs should be protected against unauthorised access.
Integrity	Information exchanged, services, CPSs, and communication links should be protected against unauthorised modifications or manipulations.
Availability	Information exchanged, services, CPS, and communication links should be available to authorised entities when requrested by such entities.
Authenticity	The management, the configuration, and operation of the onboard CPSs and services offered by CPSs should be performed by authorised entities.
Possession and Control	Information exchanged and communication links between CPSs and services offered by CPSs should be protected against the possibility that confidential data be possessed or controlled by unauthorised entities.
Utility	Information exchanged and communication links between CPSs and services offered by CPSs should be useful.

Continuation of Table 5.8	
Non-Repudiation	CPSs should not refute responsibility.
Safety Attributes	
Controllability	The ability to bring a CPS's/vessel's process into a desired state and handle hazardous events during the vessel's operations.
Observability	CPSs should be able to determine their state to enhance the situational awareness of the shore control centre.
Operability	The CPSs should be able to operate within the constraints imposed by the vessel's state.
Resilience	The CPSs should be able to absorb any disturbance caused by faults.
Survivability	The CPSs should be able to maintain the vessel's operations at some predefined acceptable level.
Graceful Degradation	The CPSs should be able to maintain possibly limited but still safe functionality.
Quality of Service	CPSs data should arrive in time and serve their purpose to perform the necessary safety functions and produce the safety messages that are needed.
Availability	The CPSs should be able to provide a stated function if demanded under given conditions over their defined lifetime.
Redundancy	The systems architecture of the C-ES should be redundant (CPSs, equipment, part and data redundancy)
Fault tolerance	The CPSs of the C-ES should continue to be operational in the event of a hardware of software failure.
Integrity	The vessel's CPSs and functions should be durable and stable.
End of Table	

Interestingly, two attributes are shared between security and safety - integrity and availability.

Shaaban et al. [68] discusses information security in the context of the cloud IoT application domain. The authors state that information security is primarily described by three main dimensions; CIA. Further distinctions are made providing

attributes application security, management of security and compliance with security standards.

5.3.1.1.5 Application Domains

Two primary papers identify the security critical application domains of IoT.

Mahbub [65] provides a considerable study on IoT security, identifying many vulnerabilities and considerations by application domain. The authors identify 12 security vulnerable applications of IoT including intelligent transport, smart cities, smart metering and more.

Security critical application domains of IoT ([65]):

- Smart Cities
- Smart Home
- Smart Environment Monitoring
- Intelligent Transportation
- Smart Metering and Smart Grid
- Industrial Automation
- Security and Emergencies
- Smart Healthcare
- Smart Retail
- Smart Agriculture

Omoniwa et al. [69] discusses application domains in regards to their research on FECIoT. They identify six practical applications for the technology which are provided below.

- Intelligent Transportation Systems
- Smart Grid
- Smart Health-care
- Smart Homes
- Smart Environment
- Smart Cities

5.3.1.2 Topic 2: Element Design

Nine primary papers contribute new modelling elements. These papers have been categorised by their application domain - Vehicle, IoT and Other - to enable comparisons to be made between elements offered in the same domain.

In the vehicle domain, **five** primary papers propose new elements for security modelling in the vehicle application domain. Five common elements and 32 unique elements were defined by these primary papers. These elements are listed below.

List of common and unique vehicle modelling elements:

- Common Elements:
  - ECU [75] [62]
  - CANNetwork [75] [62]
  - FlexRayNetwork [75] [62]
  - LINNNetwork [75] [62]
  - VehicleNetwork [75] [62]
- Unique Elements:
  - Environment [70]
  - Lidar [70]
  - Vision [70]
  - Vehicles [70]
  - GPS [70]
  - Driver [70]
  - Controller [70]
  - Web [70]
  - HTTP [70]
  - Voice [70]
  - Motor [70]
  - GPS Server [70]
  - GSM Protocol [70]
  - Firmware [75]
  - GatewayNetwork [75]
  - SoftwareProduct [62]
  - Dataflow [62]
  - Protocol [62]
  - MOSTNetwork [62]
  - GeoLocation [60]
  - UntrustedNetwork [60]
  - Vulnerability [76]
  - Threat [76]
  - Attack [76]
  - Security Incident [76]
  - Fault [76]
  - Error [76]
  - Failure [76]
  - Hazard [76]
  - Incident [76]
  - Accident [76]
  - Menace [76]

Ouchani and Khaled [70] created a simulation language for analysing cyber-physical systems and their threats, specifically in regards to the AV application domain. To achieve this they first present a meta model of the CPS domain, identifying six generalised elements - entity, object, device, social actor, protocol and server (Figure 5.8). Through this meta-model the authors created an instantiation of an AV, including the elements listed in Table 5.3.1.2. Aside from these models the authors also define other holistic aspects such as the operating environment, threat environment, countermeasure environment, and attack behaviour.

Katsikeas et al. [75] provide a new modelling language, VehicleLang, which describes probabilistic models of cyber-attacks in vehicles. This language is based off previous work done on the Meta Attack Language (MAL, [77]) which provides a meta-language and attack logic for cyber attacks. The authors extend this into a domain specific language (DSL) for the simulation of known attacks on connected vehicles. When doing so the authors identify aspects and elements that are needed

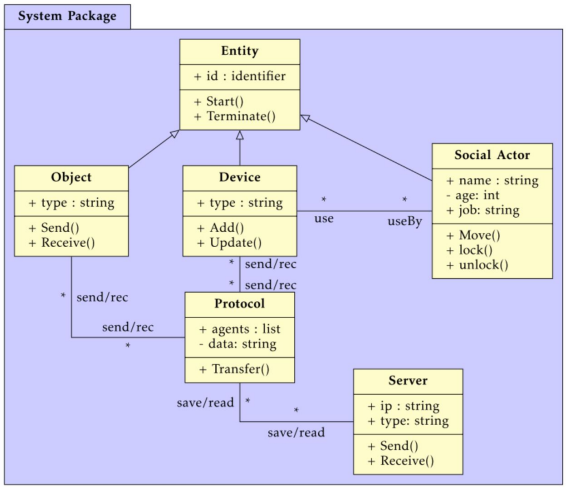


Figure 5.8: CPS Meta-Model (extracted from [70])

in order to accurately and fully describe a connected vehicle when identifying attack vectors (Figure 5.9).

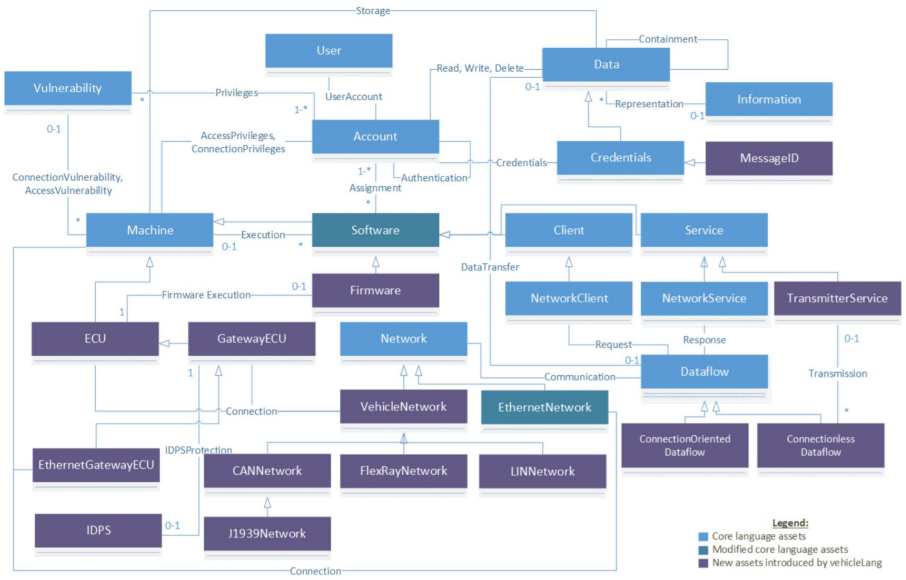


Figure 5.9: Model of VehicleLang (extracted from [75])

Xiong and Lagerström [60] extended VehicleLang, developing a privacy analysis extension for the modelling language. To do so the authors also define data types stored and accessed by connected vehicles. Four overarching categories were identified; geolocation, vehicular sensor data, biometrics data, and behavioural data. Further, through the specific extension of vehicleLang into the privacy domain, privacy specific modelling elements can be identified. The authors identify the need for two additional elements to VehicleLang; Geolocation and UntrustedNetworkService. Geolocation represents geolocation data, which implicitly identifies this data category as the most at risk for privacy concerns as the other three data types were

not instantiated within the new extension. UntrustedNetworkService identifies the fact that networks may not always be trustworthy, and by transmitting information over these networks may increase the risk to geolocation data being exposed. The extension to vehicleLang and vehicleLang itself is provided in Figure 5.10.

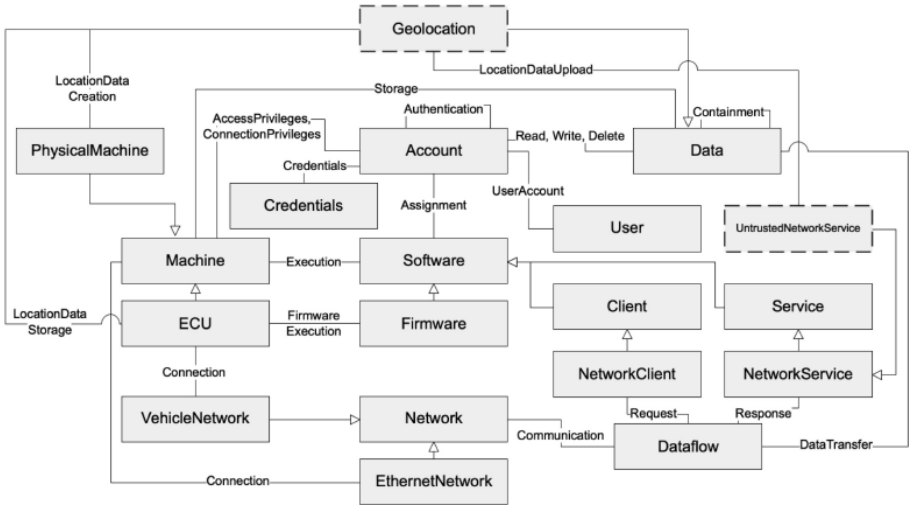


Figure 5.10: VehicleLang architecture with proposed privacy extension elements (extracted from [60])

Xiong et al. [62] model threats and attacks against connected vehicle using the tool securiCAD. SecuriCAD enables the modelling of home Local Area Networks (LANs), large corporate networks, and supervisory control and data acquisition (SCADA) systems. In securiCAD, defence strategies are selected for each component/asset. The simulation engine can then use these to illicit probabilities regarding the likelihood of attacks succeeding. The authors utilise this to provide threat modelling and in doing so provide the security critical vehicle elements. These five elements are; ECU, softwareproduct, Dataflow, protocol and network.

Two threat models were created; a threat model for the 2014 Jeep Cherokee model and a threat model for the 2015 Cadillac Escalade Model. Each were created according to their respective network typologies.

During the discussion the authors outline the importance of the firewall element in securing a connected vehicle. A keyless entry control ECU and similar assets are often entry points for attackers - access control enabled by a firewall is a strong countermeasure.

Koschuch et al. [76] discusses how safety and security are linked in the context of AV. To facilitate this the authors provide a combinatory view of safety and security best practices (Figure 5.11). The authors retrieved causation chains in regards to security and safety from various references and collated these into an initial model as shown in Figure 5.11. This model defines the elements and relations identified

between safety and security processes. The authors identify the difficulty of defining whether a security incident only leads to an error or if there are cases where a security incident could lead directly to a failure. It was decided that, while this can be contentious, all security incidents result in an error in the system and have included relation "A" to represent this.

Relation "B" identifies situations in which hazards cause a vehicle to perform an action which could be considered a threat to the security of the vehicle. The example given is if a plain text communication is sent from the vehicle in a moment of duress - enabling a threat in which information could be leaked to a bad actor.

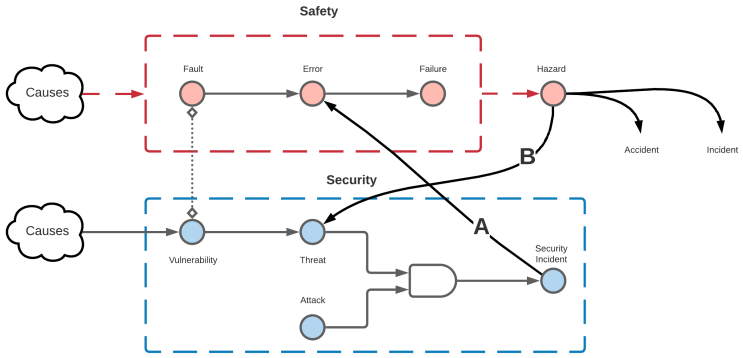


Figure 5.11: Combined safety and security view with possible causal chains (adapted from [76])

Two papers discuss element contributions in the IoT application domain. Two common elements were shared between these papers, while seven unique elements were identified. These elements are listed below (list 5.3.1.2).

- Common Elements
  - Computation [65] [78]
  - Sensing [65] [78]
- Unique Elements
  - Actuator [65]
  - Storage [65]
  - User Interface [65]
  - Device [65]
  - Software Component [65]
  - Identification [78]
  - Communication [78]

Mahbub [65] studies IoT protocols, vulnerabilities and preemptive architectonics and offers some classification regarding the structure and elements that constitute

an IoT instantiation. Four elements are classified; *identification*, *sensing*, *communication*, and *computation*.

Identification outlines how IoT technologies recognise themselves and how other entities can identify IoT devices. Many methods have been identified such as electronic item codes and uCodes. An IoT device can be identified through a tag such as Temp1 - which identified the node as a particular temperature sensor as well as identifies its position in the network.

Sensing describes how IoT devices are able to detect information within their allotted network and the transmission of this information to a data centre. Sensing IoT devices can be utilised in a wide spectrum of domains such as actuators, wearable detecting gadgets and audio sensing.

Communication underpins IoT's ability to work alongside heterogeneous applications in order to communicate specific cognitive resources. Communication standards often associated with IoT technologies are Bluetooth, IEEE 802.11, IEEE 802.15.4, Z-Wave and Long Term Evolution Advanced (LTE-A). Further, communication technology such as Radio Frequency Identification (RFID), Near Field Communication (NFC) and Ultra Wide Band (UWB) have been developed for use in IoT.

Finally, *computation* describes the computational component of IoT devices. Raspberry PI, Arduinos, Gadgeteer, and T-Mote Sky technologies have all been used to operate IoT technologies.

Zimmermann et al. [78] developed a model extraction and integration approach for EA viewpoints, models, standards, frameworks and tools in the context of IoT. A meta-model of the IoT domain is provided which defines common elements in IoT architecture - sensors, actuators, storage and user interfaces. IoT resources and their associated physical devices are differentiated in the context of locations and regions providing more explanatory power.

Finally, **two** papers are applicable in the general domain and discuss distributed control systems modelling and security visualisation respectively.

Janulevičius and Šiaudinytė [73] considers an EA analysis method for diagnosing security considerations in distributed systems, see *Security Quality Attributes* in Section 5.3.1.1 for more detail. The authors identify a distributed control systems model from the perspective of an EA technical layer (Figure 5.12). Using this model, the authors can elicit the security issues born from the inter-communication between entities via networks.

Latvala et al. [79] introduce The Metrics Visualisation System (MVS) - a visual modelling tool specifically used to facilitate the perception of security metrics and measurements. The tool operates in two stages - first, the designing of security

systems and second, the monitoring of these systems. The authors discuss visualisation techniques such as using colours to indicate changes in the value of some security metrics, guiding the user to place more attention in this area. They also experimented with element design, identifying shapes that were easily distinguishable from each other to promote visual immediacy. An example of the visual modelling technique is shown in Figure 5.13.

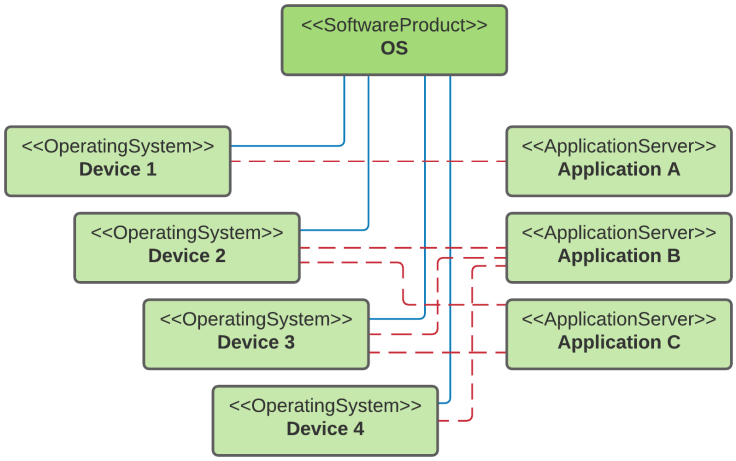


Figure 5.12: Distributed Control Systems Model (adapted from [73])

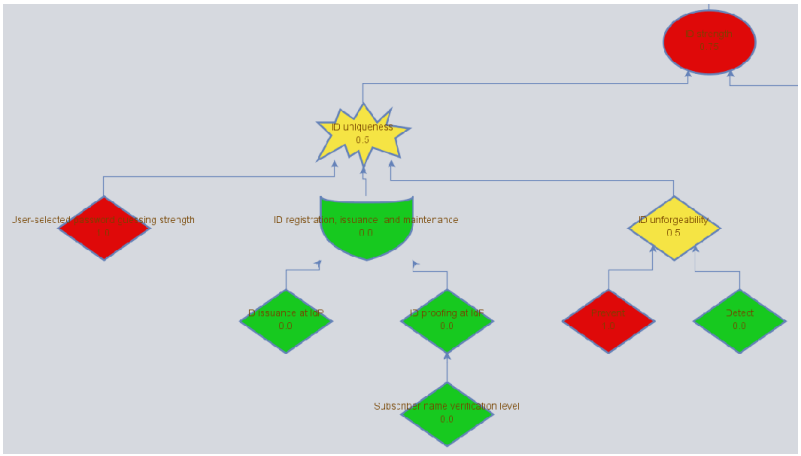


Figure 5.13: Example of MVS modelling (extracted from [79])

5.3.1.3 Topic 3: Overall Design Factors

The **15** primary papers which contribute to this sub-contribution are best described with two categorisations:

1. Layers of Frameworks
2. Design Objectives and Considerations

5.3.1.3.1 Layers of Frameworks

Six primary papers provide discussions on frameworks and layers.

Frameworks and the layers contained within them provide information on the domains technological structure and abstraction levels providing another aspect to educate potential element selection and design.

All six primary papers provide examples of IoT application domain frameworks and their associated layers. In total, 16 IoT frameworks were identified with seven of these being three layer architectures (Table 5.9), five being four layer architectures (Table 5.10), two being five layer architectures (Table 5.11) and finally two being seven layer architectures (Table 5.12).

One primary paper provides nine different CPS application domain frameworks. These constitute three, three layer architectures (Table 5.13), two four layer architectures (Table 5.14), three five layer architectures (Table 5.15) and one six layer architectures (Table 5.16)

Table 5.9: Three layer IoT architectures

Referencing Article	Layer 1	Layer 2	Layer 3
[80] from [46]	Perception	Network	Application
[81] from [69]	Sensing	Network	Application
[82] from [69]	Sensing	Network	Application
[83] from [69]	Sensing	Network	Application
[84] from [65]	Sensing	Network	Application
[80] from [65]	Sensing	Network	Application
[85] from [65]	Sensing	Network	Application

Table 5.10: Four layer IoT architectures

Referencing Article	Layer 1	Layer 2	Layer 3	Layer 4
[86] from [46]	Device	Network	Service Support	Application
[68]	Field	Communication	Data Processing	Cloud Computing
[87] in [69]	Sensing	Network	Service	Application
[65]	Sensing	Network	Middleware	Application
[88] in [89]	Implementation	Functional	Usage	Business

Table 5.11: Five layer IoT architectures

Referencing Article	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
[90] from [46]	Perception	Network	Middleware	Application	Business
[91] from [69]	Sensing	Network	Service	Application	Business

Table 5.12: Seven layer IoT architectures

Referencing Article	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7
[92] from [78]	Devices	Platform Abstraction	Security	Device Management	Service Life cycle management	Service Management	Application Interface
[93] from [65]	Physical Devices	Connectivity	Edge computing	Data Accumulation	Data Abstraction	Application	Collaboration

Table 5.13: Three layer CPS architectures

Referencing Article	**Layer 1**	**Layer 2**	**Layer 3**
[94] from [46]	Physical entities	Network	Application
[95] from [46]	Device and Process	Controller	Enterprise
[96] from [46]	Devices and Process	Controller	Enterprise

Table 5.14: Four layer CPS architectures

Referencing Article	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**
[97] from [46]	Device and Process	Controller	Manufacturer	Enterprise
[98] from [46]	Implementation	Functional	Usage	Business

Table 5.15: Five layer CPS architectures

Referencing Article	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**
[99] from [46]	Smart Connection	Data-to-information	Cyber	Cognition	Configuration
[100] from [46]	Technology	Data Management	Advanced Analytics	Digital Interface	Business Imperatives
[46]	Technology Integration	Data Management	Advanced Analytics	Digital Interface	Business Operations Evolution

Table 5.16: Six layer CPS architectures

Referencing Article	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**	**Layer 6**
[101] from [46]	Asset	Integration	Communication	Information	Functional	Business

Omoniwa et al. [69] examines FECIoT applications in the context of IoT. As part of this, they describe different methods of structuring the IoT architectures. Figure 5.14 identifies general IoT architectures and their layers.

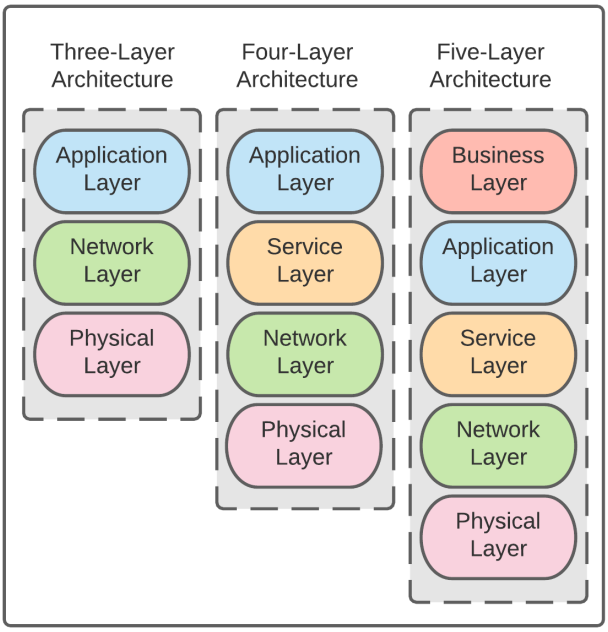


Figure 5.14: Example of MVS modelling (adapted from [69])

Three layer architectures consist of:

- Physical Layer
  - This layer is responsible for data collection, measurement and extraction of physical devices.
- Network Layer
  - This layer provides support for data to be transmitted over multiple networks and typologies.
- Application Layer
  - The layer supports the system's functionalities to the end user.

Four layer architectures provide the additional service layer:

- Service Layer
  - This layer provides a variety of services, of which can further be broken down into four categories; Service discovery, service composition, service management, and service interface.

Five layer architectures provide the additional Business Layer:

- Business Layer
  - The primary role of this layer is to record and analyse all IoT operations. In this way analytics can be carried out and models can be created which can illicit optimisation data and general benefits.

Mahbub [65] provides a detailed analysis of IoT security and while doing so provides examples of IoT frameworks and their associated layers. The author discusses how older frameworks are more generic in their expression of IoT, usually implementing a three-layer framework with no contextual aspects. More recent frameworks however, are tailored to specific application contexts [102], [103], [81].

Zimmermann et al. [78] developed a model extraction and integration approach for EA viewpoints, models, standards, frameworks and tools in the context of IoT. The authors method provides a discussion on IoT architecture, citing many frameworks and models designed for the IoT application domain. One specific architecture [104] provides a development framework for the IoT. This framework provides a set of domain-specific modelling languages and a deployment language for deployment features.

Shaaban [68] provides the CloudWoT framework, which was developed to address three key challenges. The first challenge is the structure of data within the IoT as the heterogeneity of the data in this domain makes processing this data impractical. Secondly, data formats are also heterogeneous due to the diversity of devices in IoT. Finally, IoT devices are often limited in terms of processing capacity. CloudWoT identifies four architectural layer classifications found in Table 5.15.

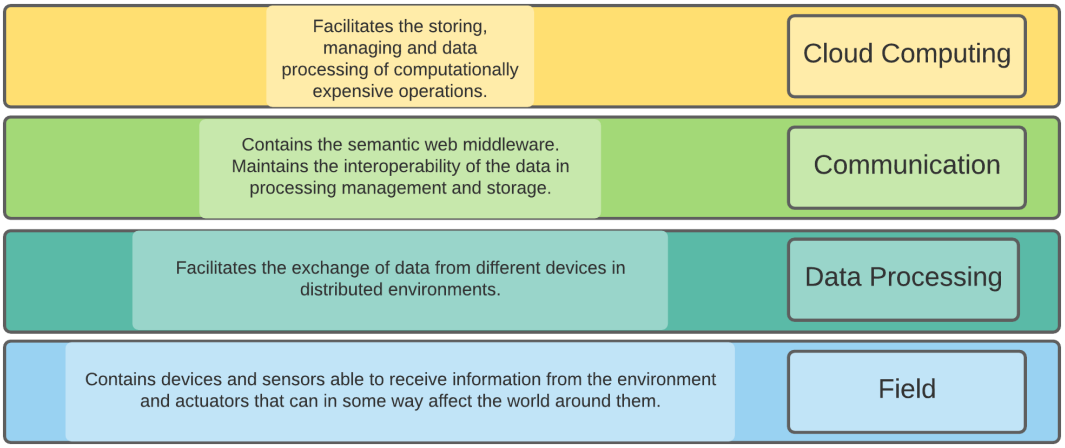


Figure 5.15: CloudWoT Framework (adapted from [68])

Baloyi and Kotzé [46] provides a data privacy model for application in the IoT and CPS domains. The authors draw upon IoT and CPS architectures to educate

their proposed solution and in doing so provide a discussion on these architectures. This discussion lends itself to identifying abstraction layers and relationships that are applicable in the modelling domain.

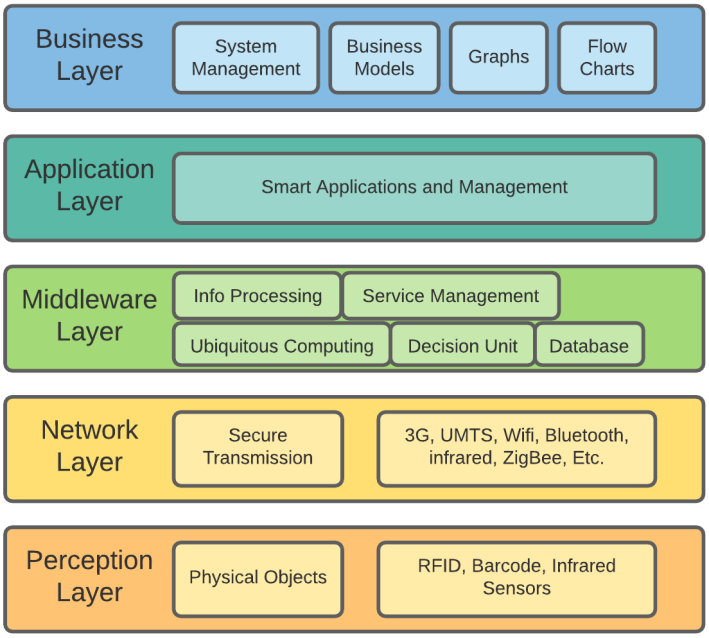


Figure 5.16: Five layered IoT architecture (adapted from [90])

Three IoT reference architectures were discussed, each with varying levels of granularity. The first architecture [105] provides three layers - perception, network and application. Next, the IoT reference model of the Telecommunication Standardisation Sector [86] contains four layers - application, service support and application support, network and device. Finally [90] provides an IoT architecture which consists of five layers - perception, network, middleware, application, and business (Figure 5.16).

CPS architectures are also discussed, analysing a range of architectures which support three to five layers. The authors conclude that CPS architectures are generally able to be characterised as a super set of IoT architectures - that CPS architectures contain all the elements required to fully realise an IoT application.

The authors make two indicative statements regarding IoT and CPS frameworks stating: "Based on our review of literature, we concur with [106] that there is currently no commonly agreed upon standard reference architecture for IoT" ([46] pg. 3) and "Although various architecture frameworks have been proposed for CPS, there is also no agreed upon standard reference architecture" ([46] pg. 4).

Morkevicius et al. [89] contributed an alignment between an Industrial Internet of Things (IIoT) framework and the unified architecture framework (UAF) to enable

traceability and interoperability in the IoT application domain. The authors based their IIoT framework instantiation from research presented in [88]. This IIoT framework consists of four layers - *implementation*, *functional*, *usage*, and, the *business* layers. These four conceptual layers were mapped to five of the offered layers in the UAF (Figure 5.17).

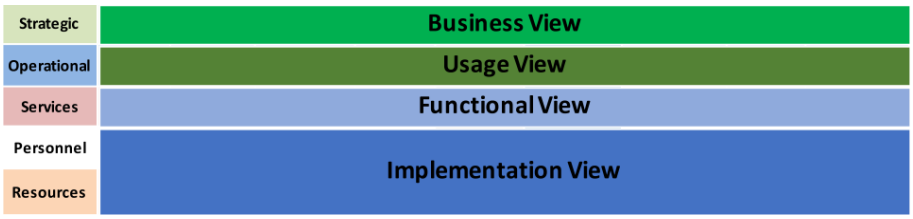


Figure 5.17: IIoT to UAF layer mapping (adapted from [89])

Five layers of the UAF were not identified within the IIoT framework - the *metadata*, *security*, *projects*, *standards*, and, *actual resources* layers. The authors identify the inclusion of the security layer as a major advantage of this joint approach, as many IIoT frameworks do not address security separately from other system components.

5.3.1.3.2 Design Objectives and Considerations

Six primary papers identify design objectives and considerations in the context of modelling.

Kavallieratos et al. [107] provide a comprehensive review of cybersecurity and safety co-engineering methods, extracting insight into the strategies used to execute these methods. From the 68 methods reviewed, 52 of these methods were found to use a form of modelling. 23 of the 52 methods provided graphical modelling capabilities while 18 of the 52 employed formal modelling. The last 11 methods used a combination of graphical and formal modelling. The popularity that modelling has seen in this domain was attributed to the ability to scale up complex systems and being able to represent different aspects related to safety and security with different viewpoints and levels of detail.

Boucké and Holvoet [108] discuss crosscutting elements and their problematic nature in architectural descriptions. Crosscutting elements of an architecture are elements which cut across multiple layers of the architecture. An example of this can be found in [68] in which the proposed security architectural concern is crosscutting the IoT architecture layers (Figure 5.18).

Another example is provided in [63] where during a discussion of AV design patterns the authors find that AV systems are highly impacted by cross-cutting

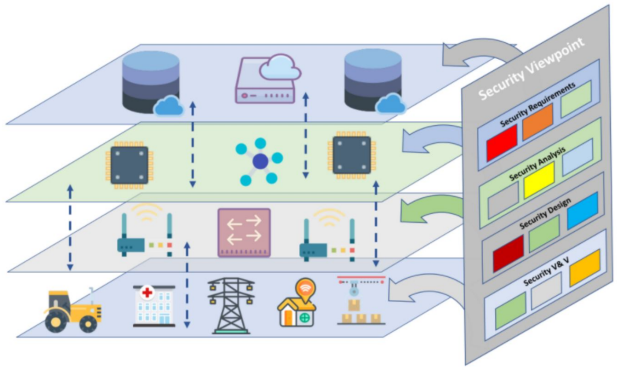


Figure 5.18: CloudWoT architecture with crosscutting security architectural concern (extracted from [108])

cybersecurity elements. The negative consequences of crosscutting elements offered by the authors are as follows:

- Since no single architectural view, or a set of views is identifiable as the description of the architectural driver, advantages of distinct design and development are lost.
- Before being able to update the design for a particular architectural driver, an architect must review all views because there are few guidelines on where to search, preventing traceability from architectural drivers to architectural elements.
- The standard notion of views does not allow explicit definitions of "open spots" (like abstract classes or parameters) that should be filled later - hampering reuse or architectural design in other applications.

Security is often a crosscutting element of architectural descriptions because the concept of security is applicable to many different technological layers.

Boucké and Holvoet [108] also discuss the underlying principles and design process behind architectural descriptions. Views, utilised in multi-view modelling and multi-view architecture, are the method that is employed to handle complexity in an architecture by improving the separation of concerns. These views are related to architectural "drivers" - the motivation behind how the architect wants to describe the architecture.

For example, a simple architectural description of IoT contains three layers; sensing, network and application. An architects driving motivation could be to contextualise this in the banking domain and attempts to include the architectural concern of security. Security is a crosscutting element of these three provided layers, and by including it, the architect has caused misalignment between the architectural drivers and architectural views. In this way complexity is added to the previous IoT architecture.

Yashchyshyn [24] examines aspects of heterogeneous modelling and methods that have been used to deal with a diversity of requirements. The author discusses complexity of design languages and how, intrinsically, a simple design language with fewer elements is desirable. There are benefits, however, to more complex notations - providing redundancy, flexibility, and diversity. The author also discusses types of modelling, including multi-view modelling, amorphous heterogeneity modelling and hierarchical multi-modelling.

Latvala et al. [79] introduces the MVS. The tool operates in two stages - first, the designing of security systems and second, the monitoring of these systems. A hierarchical graphing technique is used to provide distinction between abstract concepts such as authentication and confidentiality, as well as concrete security metrics such as number of failed attempted password inputs. During the design phase of this project the authors developed a list of design requirements of which the MVS tool would achieve.

The nine requirements were:

1. It should be easy for the user to shift from design-time usage of the MVS to the run-time useage and vice versa.
2. The core of the MVS should be separate from the model, the visualisation, and other possible modules.
3. The MVS should work on multiple platforms.
4. The MVS should help enhance the meaningfulness of security metrics.
5. The MVS should help enhance the situational awareness of the user during its run-time.
6. Calculating the value for a node from the values of its child nodes should be scriptable.
7. Node values and types should have intuitive visualisations.
8. The model used in the MVS should be saved in an Resource Description Framework (RDF) compliant format.
9. Open source third party components are preferred.

Requirements three through nine are directly applicable to the modelling aspects of MVS and identify the thought process behind the chosen modelling method. The authors also discuss how modality was achieved with their tool, allowing users to create their own extensions providing flexibility.

Kulik et al. [71] create a framework for threat-driven cybersecurity verification of IoT architectures. To do this a tool named “Alloy Analyser” is used to provide modelling and analysis capabilities in the new domain. An alloy model is able to be expressed at three distinct levels of abstraction. The highest level of abstraction provides a means to establish an overview of the system. The next level of abstraction

provides the models set theory and finally the lowest level of abstraction provides the models atoms and relations.

5.3.2 RQ3: What security strategies and mitigation techniques do micromobility companies currently employ?

To provide distinct contributions to RQ3, three topics were identified. Each topic is further decomposed into its associated components, which were developed to provide meaningful contributions to the RQ.

These topics and their components are as follows:

1. Privacy
    - Methods of Privacy
    - Technical Solutions of Privacy
2. Security
    - Methods of Security
    - Technical Solutions of Security
3. Industry Standards
    - Security Standards
    - Joint Security and Safety Standards
    - Safety Standards
    - Architecture Standards
    - Other Standards

These topics, and their components, were educated by primary paper contributions, and contain an aggregated contribution and general discussion of each contributing primary.

5.3.2.1 Topic 1: Privacy

5.3.2.1.1 Methods of Privacy

Of the **five** primary papers contributing to privacy methods, **four** identify methods for application in the CPS/IoT domain, and **one** identifies methods in the vehicle domain.

Three of the four papers which provide methods of privacy in the CPS/IoT application domain are authored by Baloyi and Kotzé in 2018, 2019 and 2020. These are presented in chronological order below.

Baloyi and Kotzé [46] developed a consolidated technical architecture by identifying similar behaviours and layers between IoT and CPS technical architectures.

The authors note that IoT architectures can be instantiated via CPS architectures, effectively making CPS architectures meta-architectures of their IoT counterparts. The proposed consolidated reference architecture is shown in Figure 5.19.

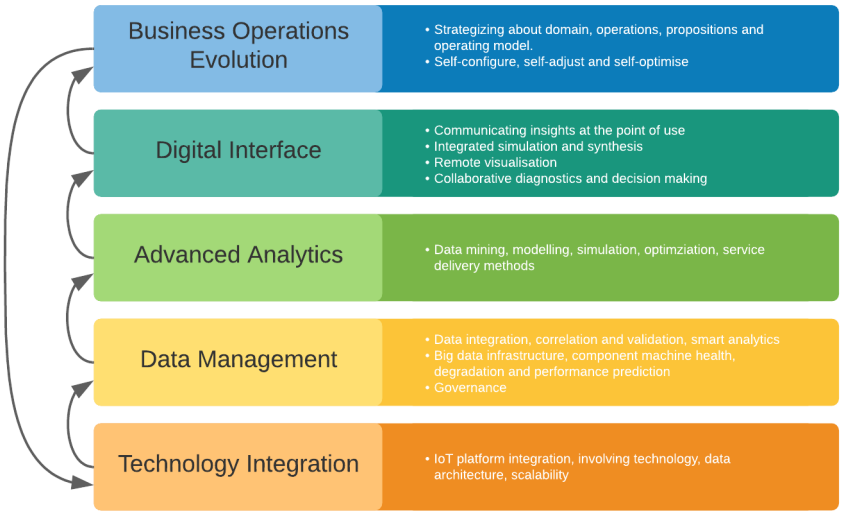


Figure 5.19: IoT and CPS consolidated technical architecture. Adapted from [46]

Using this consolidated architecture the authors propose the ICAMP model. This privacy model contains the layers from the consolidated architecture showed in Figure 5.19, however, the business operations evolution layer is divided into two complementary layers; the business imperatives layer and the self-adaptation layer. These layers are further cross-cut by six viewpoints; Who, What, When, Where, Why and How (5W1H). Privacy is enforced in the business imperatives layer, giving the CEO - or acting authority - oversight responsibilities. Further, process privacy compliance and Data privacy compliance are addressed in the self-adaptation layer and advanced analytics/data management layers respectively.

Baloyi and Kotzé [57] provides a discussion on organisational compliance in specific regards to the African Protection of Personal Information Act (APPI). They created a set of guidelines which identify the technical, organisational and legal requirements when operating under this act. A total of 43 data privacy legal compliance guidelines was defined:

1. Define purpose of collecting and processing personal information
2. Identify and categorise personal information to be or which is currently collected or processed
3. Determine if children's personal information will be collected or processed
4. Determine if you will use data subject's financial account details
5. Establish whether you conduct exception-based processing or are a responsible party with exceptions
6. Determine means of collecting and processing personal information

7. Determine or establish legal grounds for collecting and processing personal information
8. Determine whether there are applicable exclusions
9. Determine whether there are required prior authorisations
10. Determine whether there are general authorisations for special personal information
11. Determine whether there are specific authorisations for special personal information
12. Identify parties that personal information may be shared with
13. Determine whether you will be considered a responsible party, joint responsible party or operator in respect of each processing activity or piece of personal information
14. Determine whether there will be third parties exposed to or dealing with personal information or third-party systems processing or aiming to collect and process personal information
15. Determine if there will be any profiling or automated decision-making
16. Conduct privacy risk or impact assessments
17. Define appropriate security safeguards for all personal information, identified risks and processing activities
18. Ensure that there are contracts to govern the data privacy relationship with operators and third parties
19. Ensure that third party hardware and software do not prejudice data privacy
20. Ensure that personal information is used for specified or compatible purposes
21. Ensure that the repurposing of the processing of personal information is done lawfully
22. Ensure that personal information collected from other sources is collected lawfully
23. Establish data privacy policies and notices
24. Maintain documentation
25. Establish measures to support business continuity
26. Establish retention and disposal/decommissioning policies and procedures for personal information and devices containing personal information
27. Ensure that there are data privacy training programmes and awareness campaigns within an organisation
28. Establish measures to ensure quality of personal information
29. Define personal information flows
30. Design systems with privacy in mind
31. Configure systems with privacy in mind
32. Support user-centred data privacy
33. Ensure that there are appropriate safeguards for transborder transfers
34. Establish communication mechanisms for data privacy
35. Establish breach management processes for data privacy
36. Enable data subjects to exercise their data privacy rights
37. Establish mechanisms to locate all data subject personal information within an organisation
38. Determine which personal information to disclose on access requests
39. Authenticate data subjects when they exercise their rights
40. Define staff members' responsibilities and accountability for data privacy
41. Ensure that staff members understand their data privacy roles
42. Conduct data privacy audits

The authors then align these legal guidelines to a new set of operational guidelines. The operational guidelines provide actions and operational methods to implement the legal guidelines, allowing for a non-ambiguous understanding of whats required in relation to the APPI.

Baloyi and Kotzé [58] extend their previous work [57] by providing enterprises a set of benefits in regards to privacy compliance. 18 benefits were provided in this aspect:

1. Legal Compliance
2. Data Subject Trust and Confidence
3. Data Subject Retention
4. Public Trust
5. Consumer Trust and Confidence
6. Respect for Consumer Privacy
7. Improved Service Provision
8. Reducing Organisational Reputational Risk
9. Improved Risk Management
10. Data Privacy Risk Minimisation
11. Reduction of Complaints and Disputes
12. Public Perception of Transparent Practices
13. Reduced Risk of Collateral Intrusion
14. Regulated Data Sharing
15. Better Data Security/Protection
16. Encourage Adoption of CPSs and IoT
17. Improved Trade Relations and Investment
18. Organisational Management Efficiency

These two papers together [57] [58] provide a methodological framework for privacy compliance, mapping required actions and the benefits afforded to the complying enterprise.

Mahbub [65] provides a comprehensive review of IoT security. They also outline two threat modelling frameworks which emphasise privacy as a primary concern. The LINDDUN framework is classified as a framework for modelling privacy threats in software-oriented systems. LINDDUN identifies seven privacy threats:

- Linkability
- Identifiability
- Non-repudiation
- Detectability
- Disclosure of the information
- Unawareness
- Non-compliance

Finally, **One** paper identifies privacy methods in the vehicle domain.

Xiong and Lagerström [60] developed an extension to the vehicle modelling language “vehicleLang”. This extension provides a method of modelling privacy elements in vehicular systems to enable threat modelling of the system. During the execution of this threat modelling method three items need to be defined; the assets of the system, the relationships between assets and a set of assertions regarding the system. Once defined, the MAL compiler can be used to analyse the effect of the privacy extension on the system. Simulations can be compared by removing a defensive asset or relationship, providing an analysis of the relative risk between attacks.

5.3.2.1.2 Technical Solutions of Privacy

Six primary papers identify technical solutions to privacy risk. The extracted technical solutions can be found in Table 5.17.

Table 5.17: Privacy Technical Solutions

Domain	Technical Solution
CPS	Privacy-preservation by Noise Injection [Prevention] [64] random perturbations [64] estimation problems [64] minimising directed information in control loops [64] minimising fisher information [64]
IoT	Standard Cryptography for IoT Device Location Information Privacy [65] Secure Lightweight Cryptography for IoT [65] Identity authentication [65] Transient Storage [65] Fog Computing [65] Privacy Preserving Subset Aggregation (PPSA) [109]
Vehicle	Modified oblivious transfer [110] Local Differential Privacy (LDP) [60]
Software	Information Obscurity [42] Pseudonymous Identity [42]

One paper identifies a solution in the CPS application domain.

Chong et al. [64] develop a risk management framework for application in the CPS domain. The objective of the risk framework is to reduce the overall risk of attacks by deploying targeted risk mitigation strategies. To do this, cyber attacks were identified in terms of their likelihood and impact on a system. Respective mitigation strategies were then applied to these attacks. During this process the authors discuss the "Privacy-preservation by Noise Injection" technique. In this scheme, privacy is preserved via the addition of noise to a signal of sufficiently high variance that the de-scrambling of the initial data is difficult for an attacker. This is commonly known as differential privacy. Other methods of privacy preservation are briefly mentioned such as the use of random perturbations, estimation problems,

the act of minimising directed information in control loops and minimising fisher information in estimation.

Two papers identify technical solutions in regards to privacy in the IoT application domain.

During Mahbubs' [65] IoT literature review, the authors provided several technical solutions mitigating privacy risk. Interestingly the authors also outline some of the secondary benefits to privacy in regards to fog computing. Specifically, fog computing mitigates MITM attacks, data transit attacks, eavesdropping attacks and finally resource-constraint attacks. These attacks, specially the first three, lend themselves to violating personally identifiable information in an IoT context so mitigating them can be considered privacy risk mitigation.

Mahdikhani et al. [109] develop an aggregation scheme for use in Fog-Enhanced IoT which specifically addresses privacy issues in the area. This is reflected in one of the primary goals of the scheme which is to provide "different levels of privacy preservation in a user's query, the IoT devices' responses and the intermediate results among the fog nodes in the fog layer." Specifically, the issue the authors are solving is that of intermediate privacy - privacy of IoT nodes to fog nodes and fog nodes to the user. For example, the user wishes to receive some information from a subset of IoT nodes. Their request, in this scheme, is encrypted with the Paillier cryptosystem and by utilising the unique aspects of this cryptosystem this request is indistinguishable under plaintext attack.

Two papers provide technical solutions to privacy in the vehicle application domain.

Ming and Yu [110] present a prototype method which provides privacy and elevated efficiency for vehicles which query databases regarding road conditions on various routes. The authors define seven design goals of which the proposed data sharing scheme should meet; authentication and data integrity, confidentiality, location privacy preservation, identify privacy preservation, traceability, unlinkability, and Resistance to attacks. To achieve these goals the authors utilise super-increasing sequences (sequences where each number is greater than the sum of all numbers before it) providing superior efficiency and the modified oblivious transfer method - providing privacy.

Xiong and Lagerström [60] During the design of this extension the authors propose LDP. LDP is a privacy mechanism that provides a method for vehicles to defend their sensitive geolocation data, protecting privacy information of drivers. To achieve this, vehicles are able to introduce a type of noise into their data - protecting the confidentiality of the information during transit.

Finally, **one** paper provides technical solutions to privacy in the software application domain.

Hafiz et al. [42] identified two primary security patterns designed to mitigate attacks on privacy in software. *Information Obscurity*, generally, identifies methods in which information is hidden through zero knowledge communication and other methods. The second pattern is *Pseudonymous Identity* in which anonymity is achieved, providing confidentiality of an individuals data.

5.3.2.2 Topic 2: Security

In total **16** primary papers provide methods and mitigation strategies for security.

5.3.2.2.1 Methods of Security

Two papers provide security methods in the CPS application domain. Both of these papers where written by Kavallieratos, Katsikas and Gkioulos in 2020 and involve safety and security co-engineering.

Kavallieratos et al. [107] provide an in-depth survey on cybersecurity and safety co-engineering methods. These methods generally fit into three categories; security-informed safety approaches, safety-informed security approaches, and combined safety and security approaches. Each of these categories identify the primary objective of the approach - security or safety, or both. In total the authors identified nine co-engineering methods (Table 5.18) with their survey.

Table 5.18: Methods of security and safety co-engineering (extracted from [107])

Begin of Table		
Method	**Reference**	**Def**
US^2	[67]	Analysis safety hazards and security threats for CPSs in automotive vehicles through a quantitative scheme.
STPA and Six Step Model	[111]	Analysis of safety and security issues and artefacts for AVs.
FACT	[112] [113]	Failure-Attack-CounTermeasure provides a graphical approach for analysis of safety and security in the CPS domain.
CRAF	[114]	The Cyber RA Framework provides a method to elicit how a loss of data security can affect safety.
UFoI-E	[115]	The Uncontrolled Flows of information and Energy method provides a diagrammatic representation of CPS systems for risk analysis.

Continuation of Table 5.18		
AVES	[116]	The Automated Vehicles Safety and Security Analysis Framework creates four relationship matrices and a cybersecurity deployment model. The method leverages these to provide insight into safety and security concerns.
CPS master diagram	[117]	The CPS master diagram is a hierarchical three-layer representation of the studied system in different process types.
IoT medical devices	[118]	This method specifically identifies security threats that would force IoT medical devices to violate their functional safety
SARA	[119]	The Security Automotive Risk Analysis provides a method of threat modelling and RA for driver-less vehicles.
End of Table		

Next, Kavallieratos et al. [74] provide their own safety and security elicitation framework offering a method based upon the Secure Tropos and Systems Theoretic Process Approach (STPA) methods (see Figure 5.20). SafeSecTropos addresses five key limitations with other safety/security co-engineering methods:

- Objectives-driven method
  - Other methods do not elicit requirements based on safety and security objectives.
- System Models
  - Safety and Security models differ greatly in their representation, SafeSecTropos provides one model for use in both domains.
- Documentation
  - Documentation between safety and security methods differ greatly, SafeSecTropos provides interoperability between documentation.
- Conflict Resolution
  - In SafeSecTropos, each requirement can be traced back to the objectives and goals that generated it, providing a resolution mechanism for potential conflicts between safety and security.
- Representation of complex systems
  - By combining the geographical aspects of Secure Tropos and the systematic perspective of STPA, SafeSecTropos is able to present and analyse complex and interdependent systems.

Seven papers provide security methods in the IoT application domain.

As well as providing privacy frameworks, Mahbub [65] also provides two general security frameworks used in the IoT application domain. The STRIDE framework

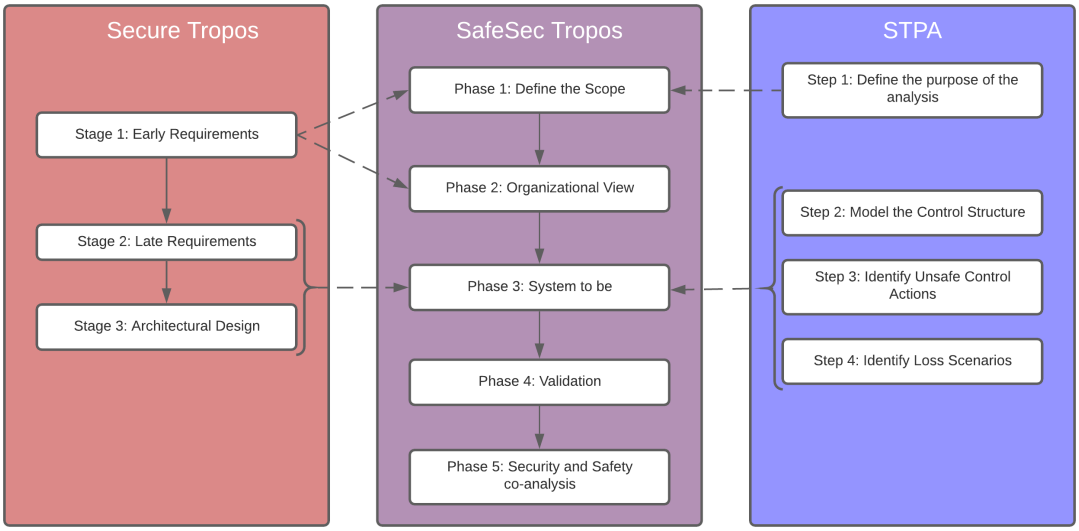


Figure 5.20: SafeSecTropos method consolidated from the Secure Tropos and STPA methods. (Adapted from [74])

has been leveraged in the IoT application domain to provide a threat model ([120] from [65]). STRIDE is a Microsoft-developed program for Security Development Lifecycle (SDL) and stands for:

- Spoofing
- Tampering
- Repudiation
- Information Disclosure
- Denial of Service
- Elevation of Privilege

The second framework described is the CORAS framework; This framework is uniquely structured such that developers are treated as distinct persons, rather than a singular group. Research performed with CORAS has yielded work describing and analysing the *health* of connectivity between IoT gadgets and smartphones ([121]).

Kulik et al. [71] develop a threat driven modelling framework for application in the IoT domain. The authors build upon the modelling language *Alloy* - a language designed to describe structural constraints in software - and its analysis tool *Alloy Analyser*. *Alloy Analyser* is a tool which is used to verify models created with the *Alloy* language, checking for correctness and whether the system meets the desired properties. The authors extend the application of these tools into the IoT domain, providing a set of system actions that occur with IoT technology (see Table 5.19), designating the rules of the new system. With these behaviours defined, common attack patterns can be applied to the model, providing a method that can define possible mitigation strategies in these systems.

Table 5.19: System actions defined in [71]

Action	Parameter(s)	Description
generate(d)	*d*, data	Action representing a subsystem generating data d (i.e. reading from a sensor)
send(d,c)	*d*, data *c*, communication channel	Action a representing subsystem sending data d via the communication channel c
acquire(d, c)	*d*, data *c*, communication channel	Action representing a subsystem obtaining data d from the communication channel c
accept(d)	*d*, data	Action representing a subsystem accepting the data d sent by another subsystem
discard(d)	*d*, data	Action representing a subsystem discarding the data d sent by another subsystem
connect(c)	*c*, communication channel	Action representing a subsystem connecting to the communication channel c
disconnect(c)	*c*, communication channel	Action representing a subsystem disconnecting from the communication channel c
recover()		Action representing recovery of a subsystem from compromised to normal mode
compromise()		Action representing a subsystem becoming compromised, i.e. malicious activity is present on the subsystem

Griffy-Brown et al. [27] develop an information security framework, specifically for executives, for application in the IoT domain. To do this the authors surveyed 59 firms, from small enterprises to large enterprises, and interviewed business leaders. Risk based approaches were identified as the most prevalent security strategies used in these enterprises, with over 80% of respondents indicating that some form of risk-based approach was implemented. To this end, the authors offer an extended risk-based method (see Figure 5.21) which provides a better estimation of cost due to risk for use in budgeting.

Finally, the authors offered emerging themes drawn from their industry interviews which were further refined into three recommendations:

- Take an Extended Risk-Based Approach
  - Cyber-security best practice is to implement a holistic risk evaluation method which provides accurate budgeting constraints.
- Be Data-centric
  - The IoT application domain specifically handles vast amounts of heterogeneous data, often increasing the difficulty of handling and processing. Data indexing and protection should be of higher consequences for enterprises operating in this domain.

- Don't forget the basics
  - Follow proven security practice and reduce risk appropriately.

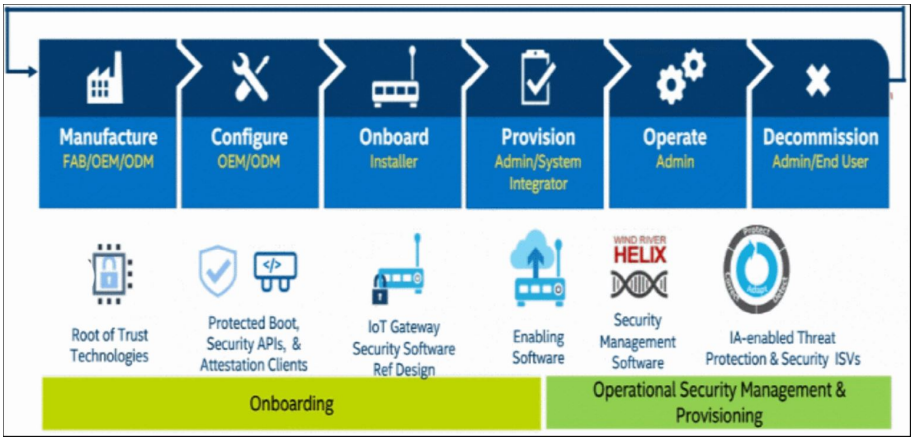


Figure 5.21: Proposed extended risk-based method (extracted from [27])

Shaaban et al. [68] present a reference model for IoT based technical solutions. During this work the authors provide some general security methods that are recommended for enterprises operating in the IoT domain. The recommended method for security in IoT is to divide the IoT system into several components or zones able to mitigate the security risks from the other parts in the system. These zones are allocated a security level, which determines the security goals and therefore the capabilities that are required to secure each zone. The authors also discuss the Microsoft threat modelling tool as a method of identifying various threats to the provided system (Figure 5.22).

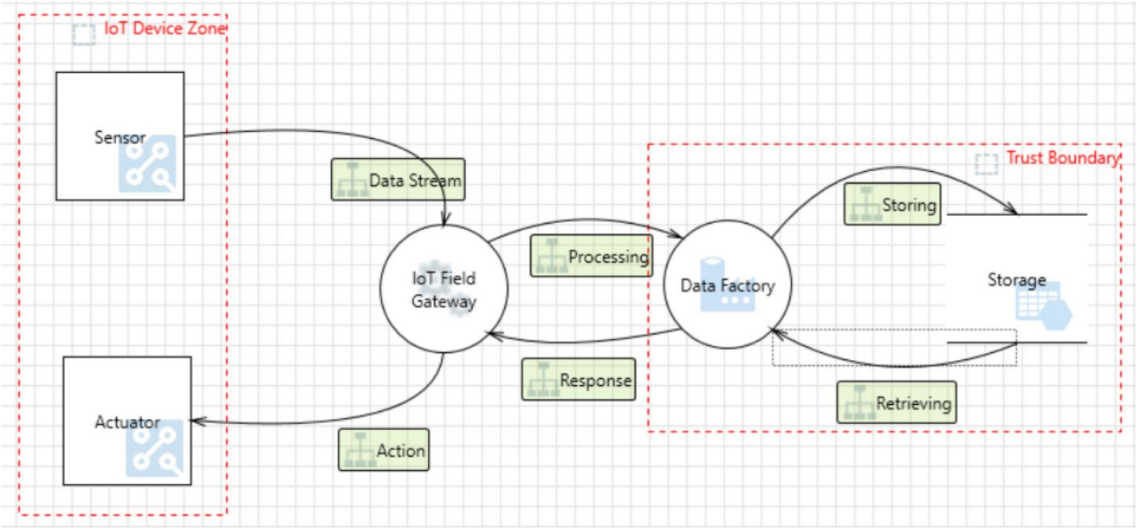


Figure 5.22: Microsoft Threat Modeling Example (Extacted from [68])

Papke [122] argues that model-based system engineering (MBSE), coupled with EA modelling languages provides a solution to agile security architecture. Agile

security, as a method, is an answer to agile threats, providing security systems that adapt to the threat landscape around them - providing flexible and timely protection against new and novel attacks. The authors argue that MBSE enables an effective implementation strategy of agile security as MBSE methods include a modelling language, modelling method and modelling tool. The combination of a semi-formal language, a defined method, and tool provides powerful benefits to the system architecture team - allowing for detailed modelling with inbuilt traceability and design consistency. This provides the engineers with the ability to modify the system and understand the implications of the modification throughout.

Augusto-Gonzalez et al. [72] present the security project *GHOST* designed to provide real time risk control for home-based IoT technology applications. During the discussion of this tool the authors provide seven security frameworks discussed in literature and implemented in industry (Table 5.20).

Table 5.20: Frameworks discussed in [72]

Framework	**Description**
IoTGuard [123]	Utilises Bro Intrusion Detection System (IDS) to detect abnormal behaviours in an IoT environment
IDS Framework [124]	Based on Anomaly Behaviour Analysis, this framework provides security for existing smart home installations.
Agent based modelling [125]	*agents* inside the smart home environment make observations and implement intended behaviour.
Heimdall [126]	A whitelist based intrusion detection technique designed for IoT devices.
Policy-based whitelisting [127]	A network layer architecture which provides mitigation implementations
Traffic filtering and anomaly detection [128]	Enforces a rule based method where every IoT device is only allowed to perform a specified behaviour.
Blockchain based smart home framework [129]	Utilising blockchain technology to provide decentralised security and privacy in a smart home environment

GHOST itself, implements a network monitoring and anomaly detection approach, providing the flexibility required from the heterogeneous IoT ecosystem. Security is compartmentalised into a five layered system architecture, enabling the independent development of each system, while preserving inter-dependency within the framework. The full architecture can be found in Figure 5.23.

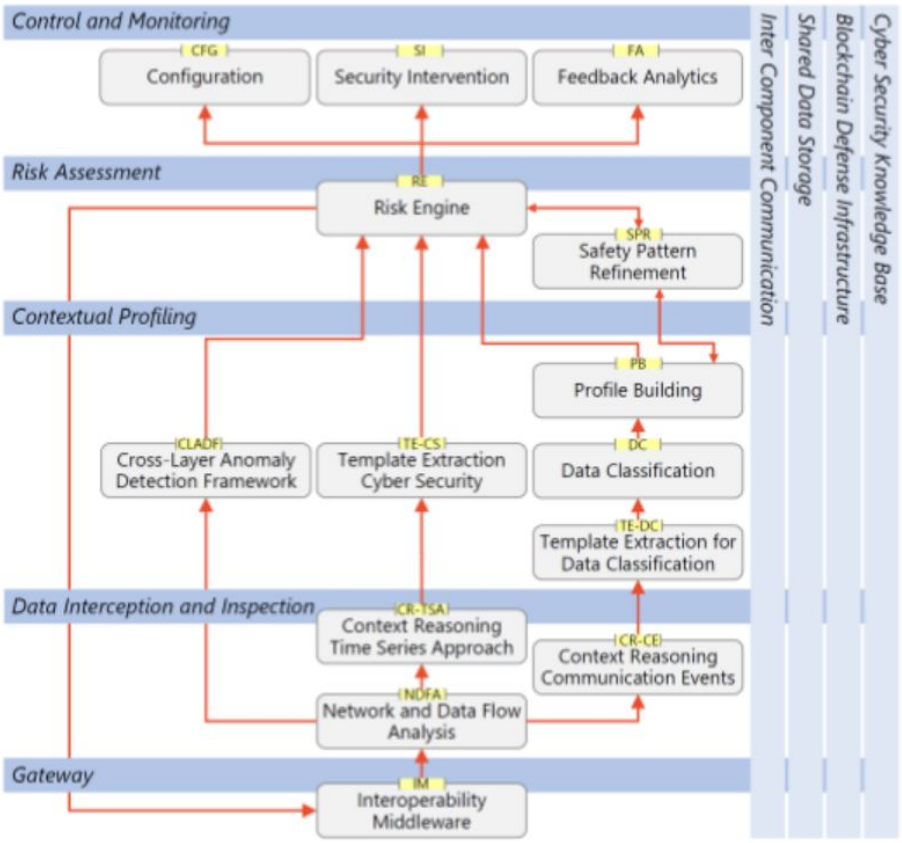


Figure 5.23: GHOST architecture (extracted from [72])

Li et al. [130] provide an improved security RA method through utilising a heuristic technique based on the Cuckoo Search (CS) algorithm. The authors provide improvements to the original CS algorithm, reducing the time needed to find an optimal solution. The authors then provide the RA method consisting of six steps. Steps one and two initialise the back-propagation neural network (BPNN) and improved CS algorithm respectively. Steps three and four process the input (risk factors) and optimise the fitness of the solution. Finally, steps five and six determine if the solution conditions are meet, and output the best solution in the set of proposed solutions.

Three papers provide security methods in the vehicle domain.

Cui et al. [67] integrate safety engineering processes (ISO 26262) with security engineering process (SAE J3061) to create a safety and security co-engineering framework. To do this a six step method was proposed which generates a *safety and security* model consisting of six hierarchies describing system functions - functions, structure, failures, attacks, safety countermeasures - as well as relationship matrices between these functions (see Figure 5.24).

The authors then provide a *Collaborative Analysis Framework* which outlines the methods required to elicit the information required by the *safety and security* model drawing from the previously mentioned industry standards (Figure 5.25). In doing

so the authors contextualise the use of the *safety and security* model in a holistic security and safety engineering method.

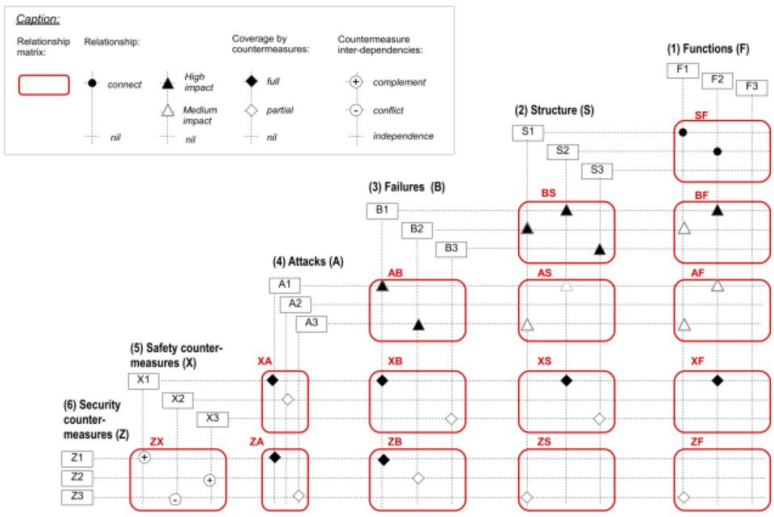


Figure 5.24: S&S model (extracted from [67])

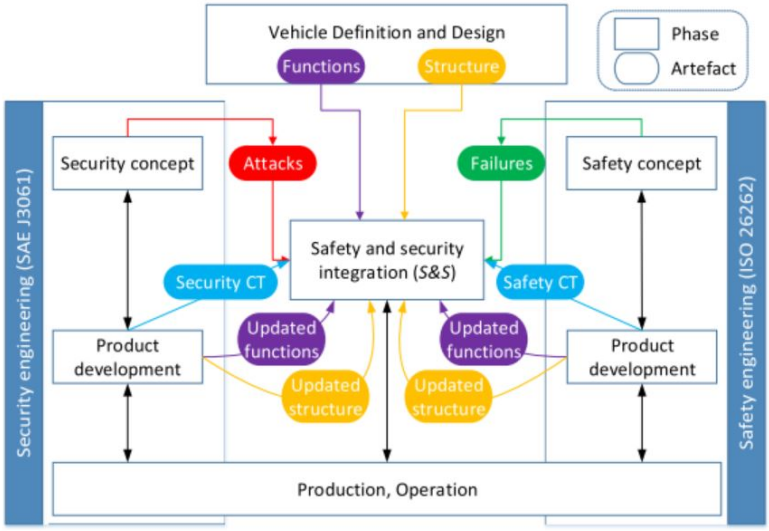


Figure 5.25: Associated Collaborative Analysis Framework (extracted from [67])

Xiong et al. [62] provide a method of simulating attacks and modelling threats in the connected vehicle domain. The authors discuss the increasing number of methods that combine threat modelling with attack simulations to provide quantitative security measurements. To implement a similar solution in the connected vehicle domain the authors utilise the threat modelling and risk management tool, *securiCAD*. Using *securiCAD*'s inbuilt simulation engine, a probabilistic attack graph can be formed containing the probabilities of a certain attack succeeding in the modelled system. By providing components of connected vehicles, security settings of these components, and possible attacks, *secureCAD* is able to produce a risk matrix

and attack path providing information on the effectiveness of in place mitigation strategies and potentially new areas of concern.

Riel et al. [131] discuss how both IT security requirements and functional safety can be included during the design of IIoT/smart devices. The method is based off two primary components - axiomatic design (AD) and signal flow analysis (SFA). Using SFA, safety goals are able to be decomposed into functional requirements (FRs) which need to be met in order for the system to meet its safety goals.

Cyber-security requirements can then be elicited through the application of a threat analysis and RA process. These security requirements, along with the safety FRs, can then be mitigated during the design process. The authors also provide a discussion on how this joint safety security process can be integrated into applicable industry standards.

Four papers provide security methods for general application over many application domains.

Islam et al. [132] explores overall security orchestration techniques through an SLR, identifying 95 primary papers. During their review the authors identified six security automation strategies and methods found in literature (see Table 5.21).

Table 5.21: Security Automation Strategy [132]

Strategy	**Description**
Auto Integration	Tools and methods that automatically connect existing security tools through APIs to streamline an incident response process.
Workflow	Workflow tools are a solution to gather and enhance alerts that automatically send instructions to analysis, auditors, and other security systems.
Scripting	Scripting tools perform actions based on custom code written by security staff, who use the scripting tools to configure existing playbook, security tools and policy.
Prioritisation	Prioritisation tools help security teams to decide critical security alerts.
Learning	Utilising Artificial Intelligence (AI) techniques and game theory models to make security systems intelligent.
Plugin & Module	Small programs of software that organisations can independently select an install based on the required configuration.

Solhaug and Seehusen [41] develop a continuous risk analysis process based on the risk analysis framework *CORAS*. The objective of this solution is to enable the continuous maintenance of risk models, lowering the overhead of this process. The proposed process is divided into two phases - the initialisation stage and the

continuous risk management stage, with each phase consisting of steps (see Figure 5.26).

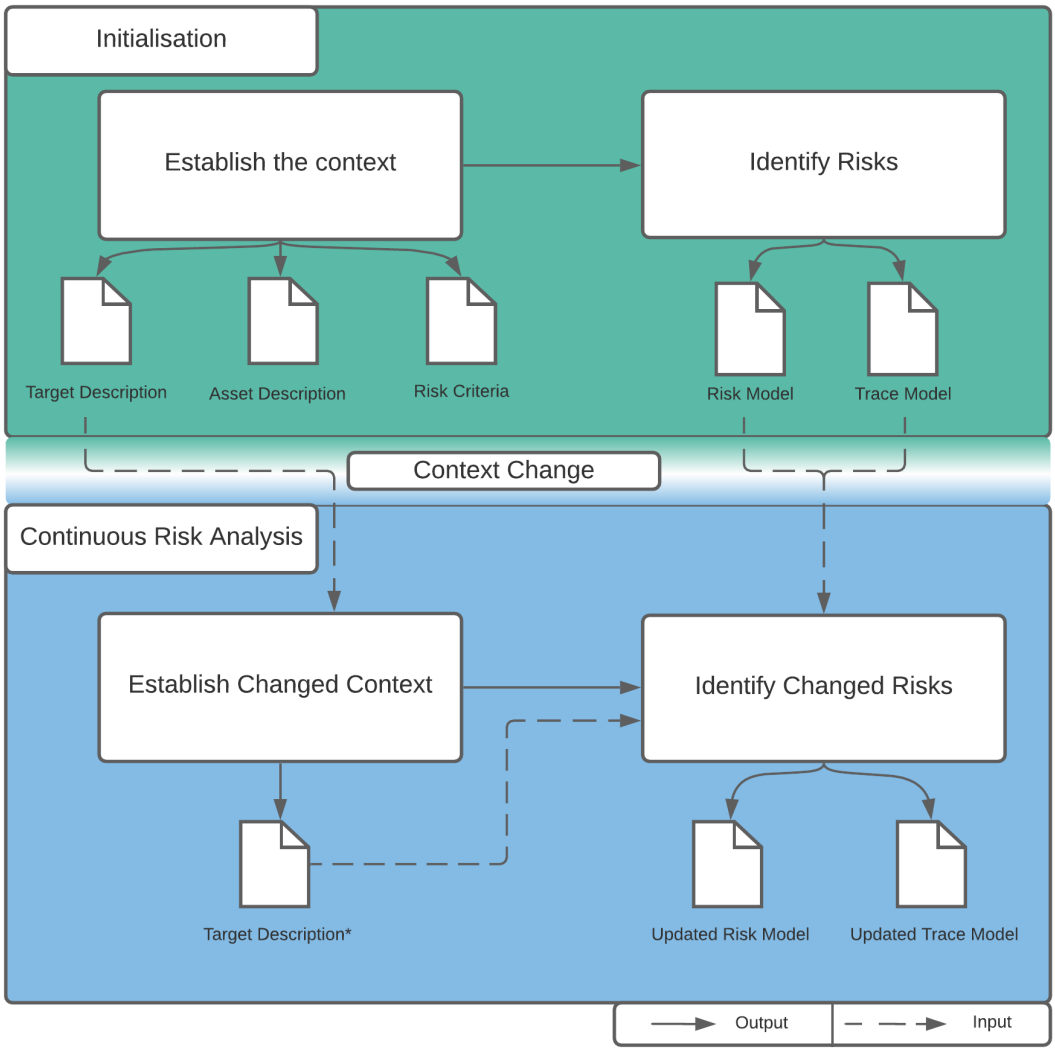


Figure 5.26: Model Driven Risk Analysis Process (Adapted from [41])

The initialisation phase consists of two steps producing five outputs. Two of these outputs are unique contributions by the authors, *Target Description* and *Trace Model*. These enable the continuous risk analysis method which is enacted during a context change. Once a context change occurs a new, updated *Target Description* is generated, which enables new risks to be calculated.

This method provides traceability from the target of analysis to its associated risks through the generated *Trace Model*'s. By doing so automated support for tracing changes and identifying risks that need to be reassessed is available.

Yigit Ozkan et al. [133] devise a method of developing customised information security focus area model (ISFAM) maturity models for Small to Medium Enterprises

(SMEs). The authors chose to base their method off of the ISFAM maturity model [134] as it is the only existing focus area maturity model (FAMM) for information security in literature. Example maturity models are provided in Table 5.22.

Table 5.22: Information and cybersecurity maturity models (extracted from [133])

Maturity Model	**Purpose/Target**
Cybersecurity Capability Maturity Model (ES-C2M2) [135]	Assessment of critical infrastructures
Open Information Security Management Maturity Model (O-ISM3) [136]	Any type of organisation
National Initiative for Cybersecurity Education – Capability Maturity Model (NICE) [137]	Workforce planning for cybersecurity
Information Security Focus Area Maturity model (ISFAM) [134]	Any type of organisation

The finalised cluster-adapted ISFAM (CA-ISFAM) creation method consisted of five steps which constitute their methodology. First, characteristic data is collected from the target SMEs. This data is used to construct a profile that represents the SME population in a cluster. Second, the frequency of individual characteristics in the SME profile are calculated. Third, these characteristics are used to create a characteristics heat map which graphically represents the data to aid in the analysis of frequencies. Forth, using the highest frequency values found in the previous steps the maximum maturity levels are calculated. These levels are then encoded into a model. Finally, a cluster adapted ISFAM model is generated after identifying how the organisational characteristics of the SMEs in a cluster affect the focus areas and the capabilities of the information security FAMM.

Ganin et al. [66] provide a risk mitigation assessment framework, analysing what countermeasures provide the most benefit - in terms of effectiveness and stakeholder interests (time, cost, complexity, maintenance) - for their associated risks. To do this the authors developed a set of criteria of which to analysis each risk against. These criteria were categorised into three groups: Threats, vulnerabilities and consequences.

Threats were found to be largely underrepresented in cyber RA models due to their complicated nature - threats require a form of probabilistic quantification which vulnerabilities and consequences do not. Vulnerabilities were found to generally be identified through a white box perspective, analysing known vulnerabilities and their effectiveness against a system. This method does not include the notion of an unknown vulnerability. Consequences in cybersecurity models were frequently described as a monetary value - that is, the cost of rectifying the situation once an event has occurred.

Using this base of information the authors developed a novel classification of threats, vulnerabilities and consequences - defining a set of criteria of which risks can be measured against. This classification is provided in Figure 5.27.

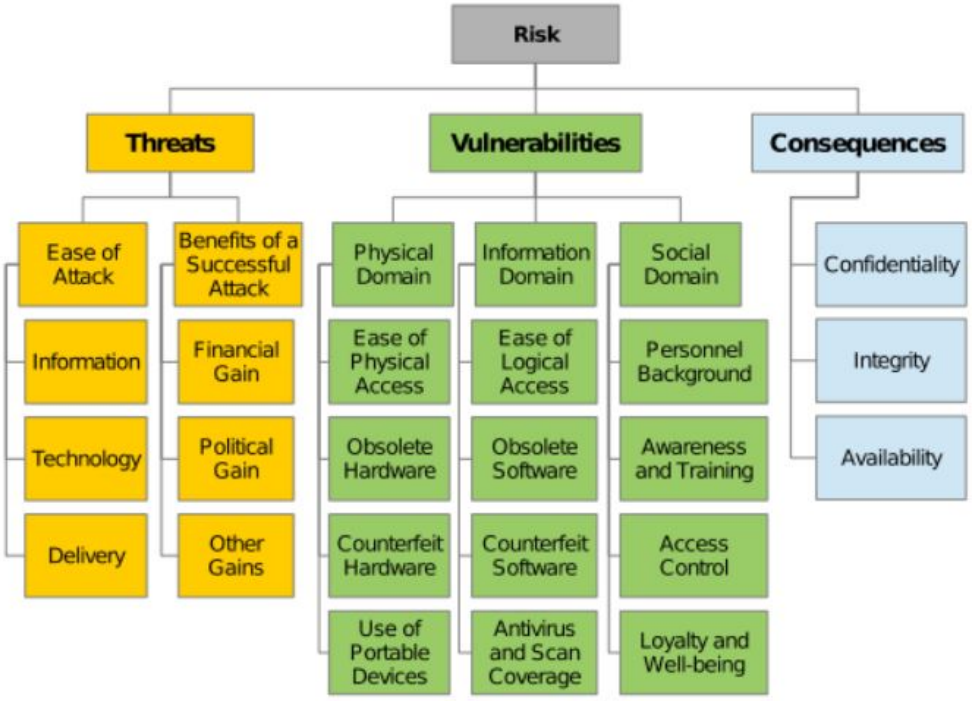


Figure 5.27: Proposed RA criteria (extracted from [66])

During the risk decision analysis these criteria are given a score either by experts or based off a set of measurable metrics. Scores assigned to the effectiveness of countermeasures are calculated using the Multi-Criteria Decision Analysis (MCDA) approach. In this way solutions and mitigation strategies are able to be compared over multiple stakeholder interests and effectiveness metrics.

5.3.2.2.2 Technical Solutions of Security

Eight primary papers contribute technical solutions and mitigation techniques to security. Primary technical solutions, grouped by their application domains, are provided in Table 5.23.

Table 5.23: Technical Mitigation techniques by application domain

Begin of Table	
Domain	**Mitigation Technique**
CPS	Tuning of Detector Thresholds [64]
	Secure State Estimation [64]
	Watermarking and Moving Target Defence [64]
	Coding and Encryption Strategies [64]
	Game-theoretic DoS protection [64]
	Event-triggered control DoS protection [64]
	Distributed Algorithms [64]
	Robust Statistics [64]
	Rational Security Allocation [64]
IoT	Secure Onboarding [65]
	Extra Interfaces [65]
	End-to-End Encryption [65]
	Firmware updates [65]
	Intrusion Detection [65]
	Datagram Transport Layer Security (DTLS) [69]
	Transport Layer Security (TLS) [69]
	Trust-management through AI, fuzzy methods, game theory and Bayesian estimation [69]
	FEC-based authentication servers [69]
	Sampling and Signature [69]
	Access Control [69]

Continuation of Table 5.23	
Vehicle	BusOffProtection [75]
	Message Confliction [75]
	Header or Timed Response [75]
	Address Space Layout Randomisation [62]
	AntiMalware [62]
	Data Extraction Prevention [62]
	Hardening [62]
	Host Firewalls [62]
	Software Patch [62]
	Static Address Resolution Protocol (ARP) Tables [62]
	APP command encryption [63]
	Dedicated Gateways [63]
	Mutual authentication [63]
General	Demilitarised Zone [42]
	Integration Reverse Proxy [42]
	Brokered Authentication [42]
	Single Access Point [42]
	Authentication Enforcer [42]
	Role Based Access Control [42]
	Compartmentalisation [42]
	Intercepting Web Agent [42]
	Security Session [42]
	MultiLevel Security [42]
End of Table	

One primary paper identifies technical solutions in the CPS domain.

Chong et al. [64] provide discussion on security in the CPS application domain elicited nine technical security solutions. Generally, these solutions are low level, theoretical solutions providing methods to detect, prevent and treat various cyber attacks. Descriptions of each technical solution are given in Table 5.24.

Table 5.24: Technical solutions identified in [64]

Technical Solution	Description
Tuning of Detector Thresholds	Anomaly detectors which raise an alarm if a received signal is sufficiently far away from nominal trajectories.
Secure State Estimation	State observers monitor the CPS system and identify how secure the current state is.
Watermarking and Moving Target Defence	CPS signals in the system are Watermarked by superimposing a globally known signal onto the carrier signal. Observers can attempt to detect the watermark, and in doing so, authenticate the signal.
Coding and Encryption Strategies	Coding an encryption strategies are employed to protect data in transit between nodes, assuring confidentiality.
Game-theoretic DoS protection	Interaction between adversary and controller is formulated as a zero-sum dynamic game. In doing so the optimal strategy for the adversary to cause a DoS attack can be formulated. From here, mitigation strategies can be implemented.
Event-triggered control DoS protection	Explicit characterisation of the adversaries frequency and duration of implementing the DoS attacks in order to adversely affect stabilisation, or consensus of the dynamical system.
Distributed Algorithms	As CPS is often decentralised, centralised mitigation strategies are ineffective. Distributed algorithms have been applied to CPS to provide mitigation techniques that are effective in this context.
Robust Statistics	Designing filters in a way which provides estimates that are insensitive to large fractions of faulty or attacked sensors - treating these as outliers.
Rational Security Allocation	Changing the system, physically or otherwise, in order to reduce the likelyhood of a cyber incident

Three primary papers identify technical solutions in the IoT domain.

During Mahbubs' [65] comprehensive review of IoT security the authors provide five technical security mechanisms. The authors discuss how fog computing, by nature, can help mitigate security vulnerabilities found within the IoT application domain. Four general cyber-attacks are mitigated through the implementation of a fog architecture, MITM attacks, data transit attacks, eavesdropping attacks, and finally, resource-constraint attacks.

As fog-computing's primary purpose is not security, it has been excluded from the general technical solutions table (Table 5.23). Descriptions of the five technical solutions can be found in Table 5.25.

Omoniwa et al. [69] provide six technical solutions during their discussion on FEC and IoT. Two of the solutions are well known, widely implemented general

solutions to internet security: DTLS and TLS. Both of these are security solutions to user datagram protocol (UDP) communication and transmission control protocol (TCP) communication respectively. The remaining four solutions are more specific to the IoT and FEC application domain. Descriptions of these solutions are given in Table 5.26.

Hernández et al. [138] discuss the limitations of security in resource constrained devices, and how secure communication can be achieved at low cost. The authors identified two primary resources whose consumption needed to be minimised - *processor speed* and *memory consumption*. Example security mechanisms that impact these resources are asymmetric cryptography, which requires further processor computation than symmetric cryptography. The authors use these, and other factors, to educate an encryption protocol mediated by a new element in the network - the *security management console*. Primarily, this solution relies on an instantiation of symmetric cryptography in which two keys, *K1* and *K2*, are agreed upon in the secure network. The redundant key allows for new key generation in the case of key exposure, providing a secure method of communication during the key generation process.

Three primary papers identify technical solutions in the vehicle domain.

Katsikeas et al. [75] provide three technical solutions during their discussion on the security modelling of vehicles. The first, *BusOffProtection*, defines how the ECUs of smart vehicles are protected by analysing incoming traffic behaviour on the CANNetwork. Next, *Message Confliction* mechanisms provide host-based Intrustion detection and prevention system (IDPS) monitors which monitor internal computing systems and network traffic. Finally, *Header or Timed Response* defends against vulnerabilities introduced by the LINNetwork protocol.

Xiong et al. [62] explore threat modelling and attack simulations in the connected vehicle application domain. The authors provide an analysis of the software *AUTORSAR* used in the selected vehicles for modelling - the 2014 Jeep Cherokee and 2015 Cadillac Escalade. This analysis provided a list of seven security mechanisms included in the software which are further described in Table 5.27.

During the discussion on *AUTOSAR*, nine other, internal, security mechanisms are described. These solutions are not included in Table 5.23 as they are not technical in nature - they more identify practices which, if done correctly, can mitigate security risk. These are given below:

- Properly Configured
  - Protects against human error
- Vendor Support
  - If the software is supported/has access to latest patches and updates
- No Patch-able Vulnerability

  - If the software has no known vulnerability
- No Un-Patchable Vulnerability
  - The software product is not inherently vulnerable
- Software Language
  - The software is written in a language which has a low error rate, and performs checks on memory
- Scrutinised
  - If the software has been thoroughly tested to ensure correct behaviour
- Secret Binary
  - Protection of the binary to make it difficult for attackers to utilise it to find new vulnerabilities
- Secret Source Code
  - If the source code is protected from unwanted scrutiny
- StaticCode Analysis
  - If there is a tool that provides pen testing of the software

Messnarz et al. [63] describes how fail operational states can be reached in AVs, and while doing so provides a list of cybersecurity design considerations, some of which are technical solutions. Three technical solutions were identified and described in Table 5.28. The authors, similarly to [62], provide non technical mitigation strategies alongside these technical solutions. In total, four non-technical solutions were identified:

- Secure development
  - Verified and secure development of the control APPs
- Secure Socket Layer (SSL) Libraries
  - The manufacturer will provide SSL libraries to be used inside the car.
- Industry best practice and security governance
  - Security requirements for access to in-vehicle data and resources and the security requirements for applications and services are based on advanced industry best practice, standards and recommendations.
- Kerckhoffs Principle
  - Security elements should be designed assuming the attacker has full access to the system.

Finally, **one** paper identified technical solutions for application in the software domain.

Hafiz et al. [42] provided a holistic security pattern language for the software domain. The authors originally planned on contributing a collation of security patterns however, they decided to utilise pattern languages as other categorisation methods did not guide practitioners in selecting the correct pattern. Pattern languages are more flexible as they do not utilise categories but rather connections between patterns – a type of ontology. The process of generating the pattern language is well documented and provides the methodology they used during the creation of the pattern language.

The pattern language describes many technical and process based security solutions in the software domain. Table 5.23 includes the proposed unified patterns (Figure 5.28) which the authors have collated. These unified patterns provide an abstracted representation of technical solutions to security.

The authors also defined a set of higher level patterns, identifying processes and methods which are used when assessing what security solutions should be implemented (Figure 5.29).

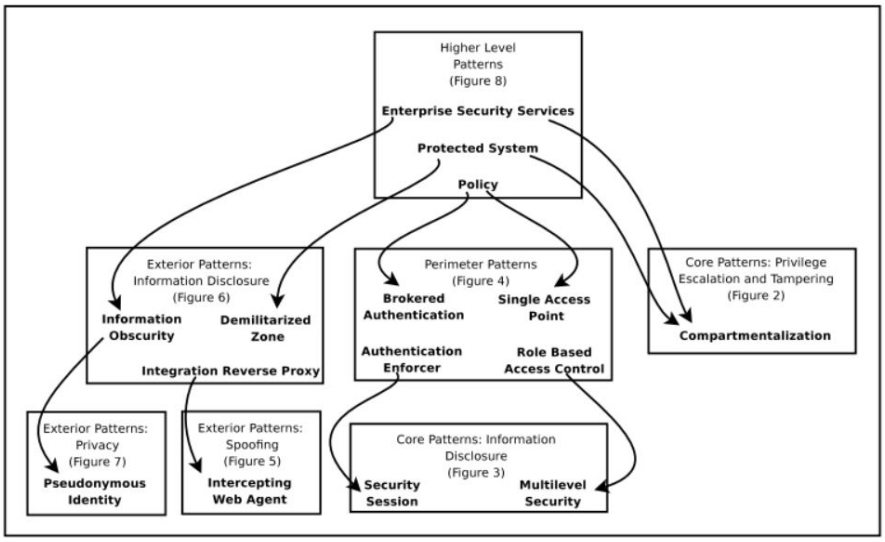


Figure 5.28: Unified Security Patterns (extracted from [42])

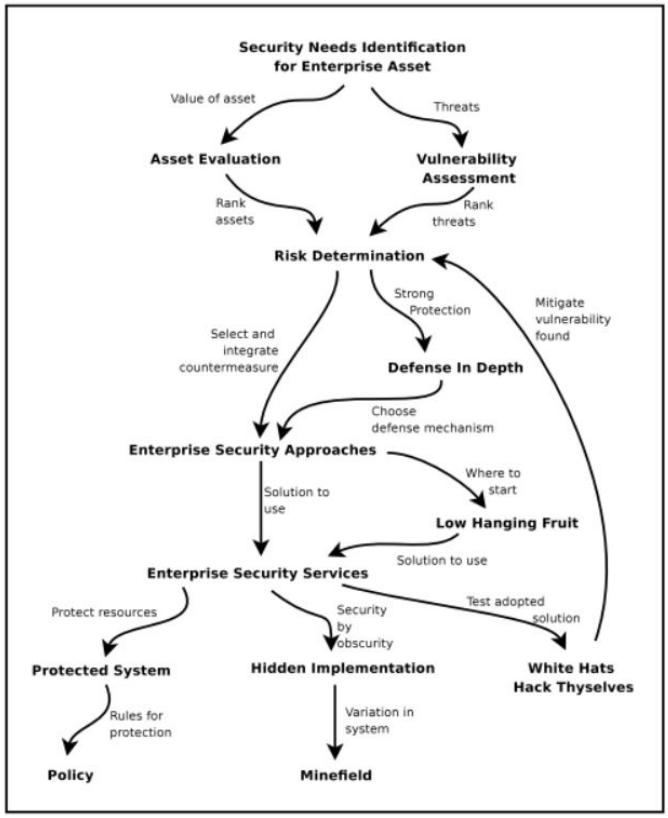


Figure 5.29: High Level Patterns (extracted from [42])

Table 5.25: Technical solutions described in [65]

Technical Solution	**Description**
Secure Onboarding	Maintaining encryption keys at the stage where another device or sensor is inserted into an IoT environment.
Extra Interfaces	Limiting the attack surface by reducing the facilities and functionalities to end-clients.
End-to-End Encryption	End-to-end protection is required in the application layer to facilitate confidentiality and integrity
Firmware updates	Due to the context IoT devices are frequently used in they often are unable to update frequently and regularly. Pipelines need to be in place to install new firmware once it is available.
Intrusion Detection	Fog nodes may recognise assaults based on city infrastructure by teaming up with their neighbouring hubs.

Table 5.26: Technical solutions described in [69]

Technical Solution	**Description**
Trust-management through AI, fuzzy methods, game theory and Bayesian estimation	Methods for managing and evaluating trust in an IoT environment
FEC-based authentication servers	Rather than centralised methods for authentication, distributed FEC authentication methods are more effective.
Sampling and Signature	Provides integrity via a local collector who acts as coordinator and periodically transmits the sampled packets to a global traffic analytic.
Access Control	Strategies and methods used to implement access control need to be lightweight as IoT devices are resource constrained.

Table 5.27: Technical solutions described in [62]

Technical Solution	**Description**
Address Space Layout Randomisation	Hardens ECUs against buffer overflow attacks
AntiMalware	Detects and removes malicious malware
Data Extraction Prevention	Protects against buffer overflow by making memory areas non-executable
Hardening	Disabling unused services, ports and hardware outlets to reduce risk area
Host Firewalls	Management of traffic shared between hosts
Software Patch	Host runs the latest security patch
Static ARP Tables	Protects against ARP table spoofing

Table 5.28: Technical solutions described [63]

Technical Solution	**Description**
APP command encryption	APP commands are encrypted, with different encryption methods used per communication media
Dedicated Gateways	Dedicated gateways are used during communication to a cloud network.
Mutual authentication	Mutual authentication to verify that the received.transmitted data sources and destinations are legitimate.

5.3.2.3 Topic 3: Industry Standards

In total **12** primary papers discussed industry standards. Each standard is enumerated with their titles and referencing articles in Table 5.29.

5.3.2.3.1 Security Standards

nine primary papers provide standards for use in the security domain. In total, **8** industry standards were identified.

ISA/IEC 62443 is a series of standards providing frameworks which address and mitigate security vulnerabilities in industrial automation and control systems (IACSs) [139]. This is discussed in [68], who create an IoT reference model, [107], who produced a survey on safety and security co-engineering methods, and, [74] who provide a method of joint safety and security requirements elicitation.

ISO 21434 provides details on cybersecurity engineering in the vehicles application domain. This standard is under development at time of writing, and will provide details on vehicle lifecycle from design to decommission in regards to cybersecurity engineering [140]. This is discussed in [63], who provide discussion on AVs and how they can achieve safe fail operational states.

ISO 27000 is a widely cited series of information security management standards which provides terms, definitions and information system security methodologies [141]. This is discussed in [68], who create an IoT reference model, [107], who produced a survey on safety and security co-engineering methods, and, [74] who provide a method of joint safety and security requirements elicitation.

ISO/IEC 27034 is a family of standards which provide guidance on information security in regards to designing, programming, implementing and utilising application systems. This enables applications to provide the necessary level of security to support organisations information security policy [142]. This is discussed in [74], who provide a method of joint safety and security requirements elicitation.

ISO/IEC 29100 provides a high-level privacy framework enabling the protection of personally identifiable information transmitted in communication and IT [143]. This is discussed in [57], who provided a set of guidelines for data privacy and compliance for enterprise.

NIST 800-30 provides guidance in regards to RA of federal information systems and organisations [144]. This is discussed in [145], which provides a SLR on safety and security co-analyses.

NIST SP 800-53 contains a catalogue of security and privacy controls for federal information systems and also, an elicitation process for selecting controls

to protect organisational operations, assets and individuals [146]. This is discussed in [76], which discusses joint modelling of safety and security in the context of autonomous driving.

SAE J3061 outlines recommended practice in regards to vehicular cybersecurity. These practices cover a wide range of vehicles from commercial to military [147]. This is discussed in [131], discussing the integration of safety and security in the IoT domain, and, in [148], who provide an analysis framework for safety and security in AVs.

5.3.2.3.2 Joint Security and Safety Standards

five primary papers provided combined security and safety standards. In total, **Five** industry standards were identified.

ISO 31000 addresses operation continuity and reassurance in terms of economic resilience, professional reputation and environmental/safety outcomes through customisable risk management solutions [149]. This is discussed in [41] in which a method of agile, model-driven risk analysis is proposed.

IEC 62645 is an industry standard providing information on nuclear power plants, specifically on their control and electrical power systems [150]. This is discussed in [107] who provide a survey on safety and security co-engineering methods.

IEC 62859 provides more information on nuclear power plants, outlining a framework designed to manage the interactions between safety and cybersecurity systems and methods [151]. This is discussed in [74], providing a method for joint safety and security requirements elicitation.

EN 50126:1999 defines the concept RAMS: Reliability, Availability, Maintainability and Safety for application in the railway domain. These concepts define requirements for all major railway systems: from major systems down to combined individual components and sub-systems (including software) [152]. This is discussed in [76], providing a joint modelling method for safety and security in the context of AVs, and [145], who provides a SLR on safety and security co-analysis techniques.

EN 50128:2011 builds upon the EN 50126 standard, specifying the process and technical requirements for developing software in the railway application domain. It provides specific emphasis on the safety implications of these systems [153]. This was discussed in [76], who provide a joint modelling method for safety and security in the context of AVs.

5.3.2.3.3 Safety Standards

Six primary papers provided safety standards. In total **six** industry standards were identified.

EN 50129:2016 outlines safety-related electronic system requirements, specifically signalling systems, in the railway application domain [154]. This is discussed in [76] who provide a joint modelling method for safety and security in the context of AVs.

ARP* 4761 describes guidelines and methods for safety assessment for the certification of civil aircraft. This is discussed in [76] who designed a joint modelling method of safety and security in the context of autonomous driving.

IEC 61508 defines basic functional safety, applicable in all industries. Functional safety is defined as "part of the overall safety relating to the EUC (Equipment Under Control) and the EUC control system which depends on the correct functioning of the electrical, electronic, or programmable electronic safety-related systems, other technology safety-related systems and external risk reduction facilities"" [155]. This is discussed in [131], outlining the integration of safety and security in the IoT domain, [107], who provides a survey of safety and security co-engineering methods, and finally, [76] who designed a joint modelling method of safety and security in the context of autonomous driving.

ISO/PAS 21448 provides guidance to engineers on the design, verification and validation measures to achieve safety of the intended functionality (SOTIF). SOTIF is defined as "the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable misuse by personnel" [156]. This is discussed in [76] who designed a joint modelling method of safety and security in the context of autonomous driving.

UK Defence Standard 00-56 sets out all safety requirements for all contractors providing products, services or systems to the Ministry of Defence (UK), and guides the "managing of Risk to Life associated with operation of military systems" [157]. This is discussed in [76], discussing joint modelling of safety and security in the context of AVs.

ISO 26262 is a widely cited standard which defines functional safety for electrical and/or electronic systems in road vehicles [158]. Six primary papers discuss this standard, all of which provide research on either the integration of safety and security, or discuss safety and security co-engineering literature. These papers are ([131] [148] [107] [63] [76] and [145])

Mil-Std 882 outlines the Department of Defence (US) systems engineering approach designed to eliminate hazards, where possible, and minimise risk where hazards cannot be eliminated [159]. This was discussed in [76], discussing joint modelling of safety and security in the context of AVs.

5.3.2.3.4 Architecture Standards

Two primary papers provided architectural standards. These papers identified **two** industry standards.

IEC/PAS 61499 provides a generic architectural description and guidelines for the use of function blocks in distributed industrial-process measurement and control systems [160]. This is discussed in [73] in which a method of security issue diagnostics integration into EA is proposed.

ISO/IEC/IEEE 42010 identifies the requirements of the description of systems, software and EA. It standardises the practice of architecture description by providing a standard conceptual foundation for these methods [161]. This is discussed in [46] in their discussion on IoT and CPS privacy.

5.3.2.3.5 Other Standards

Two primary papers provided other standards. These papers identified **three** industry standards.

ECSS-Q-ST-80C defines a set of software product assurance requirements. These requirements provide ease of maintenance for space system software [162]. This is discussed in [74], who design a joint safety and security requirements elicitation method.

ISO/IEC 25010 provides leading models for the assessment of a software product. This provides measurable metrics, establishing performance of software processes leading to improvement propositions [163]. This was discussed in [74], who design a joint safety and security requirements elicitation method.

SAE J3016 provides a classification taxonomy for vehicle automation, identifying levels of automation from full to no automation [164]. This was discussed in [148] who provided an analysis of safety and security in AVs.

Table 5.29: Referenced Standards

Begin of Table		
Standard	**Name of Standard**	**Referencing Article**
Security Standards		
IEC 62645	Nuclear power plants - Instrumentation, control and electrical power systems - Cybersecurity requirements	[107]
IEC 62859	Nuclear power plants - Instrumentation and control systems - Requirements for coordinating safety and cybersecurity	[74]
ISA/IEC 62443	Security capabilities for control system components	[68] [107] [74]
ISO 21434	Road vehicles - Cybersecurity engineering	[63]
ISO 27000	IT - Security techniques - Information security management systems	[68] [74] [76]
ISO 31000	Risk Management	[41]
ISO/IEC 27034	IT - Security techniques - Application security	[74]
ISO/IEC 29100	IT - Security techniques - Privacy framework	[57]
NIST 800-30	Guide for Conducting RAs	[145]
NIST 800-53	Security and Privacy Controls for Federal Information Systems and Organisations	[76]
SAE J3061	Cybersecurity Guidebook for Cyber-Physical Vehicle Systems	[67] [131]
Security & Safety Standards		
EN 50126:1999	Railway Applications – The specification and demonstration of RAMS	[76] [145]
EN 50128:2011	Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems	[76]
EN 50129:2016	Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling	[76]
Safety Standards		
ARP* 4761	Aerospace Recommended Practice 4781	[76]
IEC 61508	Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems	[131] [107] [76]
ISO/PAS 21448	Road vehicles - Safety of the intended functionality	[76]

Continuation of Table 5.23		
UK Defence Standard 00-56	Safety Management Requirements for Defence Systems	[76]
ISO 26262	Road vehicles - Functional safety	[67] [63] [131] [107] [76] [145]
Mil-Std 882	Department of Defence (US) Standard Practice: System Safety	[76]
Architecture Standards		
IEC/PAS 61499	Function blocks for industrial-process measurement and control systems	[73]
ISO/IEC/IEEE 42010	Architecture description	[46]
Other		
ECSS-Q-ST-80C	Software product assurance	[74]
ISO/IEC 25010	Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models	[74]
SAE J3016	Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems	[67]
End of Table		

5.3.3 RQ1: How are security aspects being incorporated into ArchiMate EA Models?

This RQ identifies primary studies which utilise ArchiMate to provide value in the area of security. In total **six** primary papers were identified. Each of these papers offer unique applications of ArchiMate and as such, no meta categorisation is made. Each paper is discussed individually, identifying the authors objectives, methods and outcomes below.

Berkel et al. [165] integrates threat analysis methods and EA modelling techniques to mitigate security challenges in the smart city domain. The authors decided upon EA modelling as it is designed around the multi-view modelling paradigm, enabling viewpoints such as data, security, governance, and business to be represented in the same model. ArchiMate, specifically, was chosen for this method as it is widely used in industry [165] and provides the RSO extension which facilitates risk modelling.

To facilitate the threat analysis of smart cities the authors provided a baseline architecture defined in ArchiMate that consists of four layers - Business Layer, Application Layer, Technology Layer and the Physical layer. Also, a five step elicitation method was applied to this architecture, identifying security requirements, controls and risk responses.

Using this process the authors produce seven risk, requirement and response models which constitute an information security architecture for smart cities. These models identify likely attack vectors and their associated security measures. These models provide stakeholders with a method to monitor security requirements and integrate these into the overall EA.

Hacks et al. [166] provides a translation method which is able to convert models developed in ArchiMate to MAL instances. The authors identify the difficulty of understanding MAL DSLs due to their code like complexity. It was found that translating ArchiMate models into MAL instances reduces its complexity, providing a platform which is widely understood. The authors propose three advantages to this method. First - tool support for ArchiMate is widely available, second - researchers have already proposed methods of security modelling in ArchiMate and, finally, EA models containing IT assets and modelling in ArchiMate can serve as input, avoiding the need to model them twice.

The solution can be divided into two distinct problems - threat modelling in ArchiMate and translation/alignment with MAL concepts. The authors decided to provide a comparatively simple threat modelling method consisting of one new element - threat, and a set of relations which are translated into *OR* and *AND* gates for the MAL translation. Once the translation is completed, attack simulations can

be run on the resulting model.

Närman et al. [167] utilise the ArchiMate language to enable availability analysis through fault trees. ArchiMate is extended to incorporate a Probabilistic Relational Model (PRM) enabling analysis to take place on the modelled system. Similarly to [166], the authors define two new classes to the ArchiMate meta-model, providing support for *AND* and *OR* gates, enabling more effective representation of fault trees and therefore availability. In total four class attributes were implemented in order to represent availability Boolean's during modelling:

- Function.Availability
- ActiveStructureElement.Availability
- ANDGate.Availability
- ORGate.Availability

Each attribute has two states - *up* or *down* with each element having a certain probability of being in either state. By embedding this information in ArchiMate models, the authors are able to predict availability of services within enterprise.

Korman et al. [35] provide an alignment of ArchiMate with various RA methods in order to identify input information for each method. The authors discuss how RA methods are often compared in terms of features offered, and contribute an analysis of the required input data and effort for each method. The chosen strategy of analysis was to model each method's input suggested in ArchiMate to afford a standardised comparison. In doing so the authors also provide an analysis of how these methods align to the concepts and elements provided by ArchiMate.

Of the twelve selected RA methods, seven were successfully mapped to the core concepts of ArchiMate (2.0). Due to the generality of ArchiMate however, these results have varying levels of confidence between reviewers regarding their correctness. For example, the researchers were able to map the RA method CORAS to ArchiMate with a confidence of 80%.

Zhi et al. [168] provides a solution to the cognitive and operational dissonance created when manipulating and interpreting independent diagrams. Traditionally, quantitative evaluation of an architecture is achieved through a DSL, distinct from the language used to describe the EA or its associated security cases. The authors unify these processes into one method which leverages ArchiMate, effectively reducing the number of distinct languages required during these processes. As ArchiMate is directly poised to model EA the authors provide new methods for the other two processes - quantitative evaluation and security assurance cases. Quantitative evaluation is achieved through integrating *Soft goal Interdependence Graphs* (SIG) into ArchiMate's respective elements, enabling the analysis of various aspects of an architecture depending on what factor is measured. Security cases are introduced by

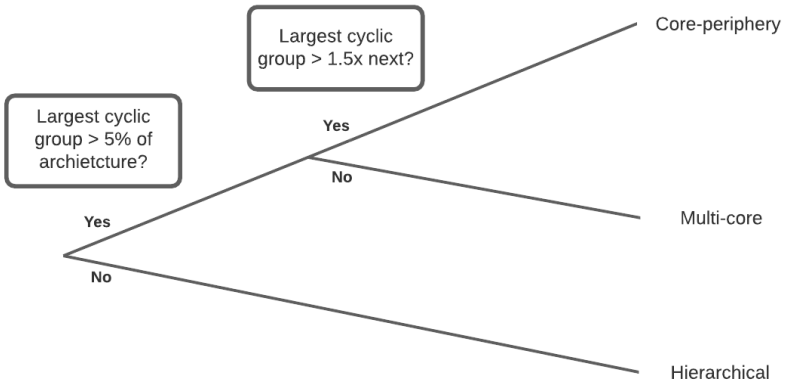


Figure 5.30: Architectural classification scheme based off cyclic groups (adapted from [170])

integrating ArchiMate with the *Intra Model Security Assurance* (IMSA) method. Drawing on previous research in the area [169] an alignment of ArchiMate's elements to IMSA's concepts is defined. Further, the authors extend this definition to include concepts relevant to SIG, enabling a joint IMSA/SIG modelling approach in ArchiMate.

Xiong et al. [170] defines a method of architecture analysis, leveraging the *Hidden Structure Method* (HSM) and *Dominator Analysis* (DA). This approach relied on transforming an architecture into a *Design Structure Matrix* (DSM) through the HSM, enabling a description of the architectures overall coupling between components and cyclic groups. Once performed, the architecture can be classified as either a core-periphery architecture - an architecture with an identifiable and substantial cyclic core, a multi-core architecture or a hierarchical architecture - an architecture with very few, small cyclic groups (Figure 5.30).

Once classified, the architecture can be re-visualised, revealing its *hidden structure* and enabling DA, a form of graph theory which outlines what elements are dominators, and therefore corner stones of the architecture. The authors utilise ArchiMate in their case study as it provides relationships and elements which can be identified by DSM from its solution architecture. Once the above process is run on an architecture, the new architecture and created output matrix can be used to provide mitigation opportunities and strategies for the next architecture iteration.

5.3.4 RQ4: What support do architectural design languages provide for security in EA?

To provide distinct contributions to RQ4, eight topics were identified with their associated components. These topics and components are as follows:

1. Attack Description
    - Attack Trees
    - Attack Graphs
2. Risk Assessment
3. Architectural Automation
4. Cybersecurity Modelling Languages
5. Other
    - Literature Review
    - Miscellaneous

Each topic was educated by primary paper contributions, and contain an aggregated contribution and general discussion and description of each primary paper.

5.3.4.1 Topic 1: Attack Description

Attack descriptions are methods of understanding and documenting how attacks are carried out, and their possible effects on the target system. They describe each variable and element that constitute an *attack path* in order to achieve their *attack goal*.

5.3.4.1.1 Attack Trees

Five primary papers provided insight into graphical attack trees and their applications.

Nagaraju et al. [171] provides a survey of fault and attack tree modelling, identifying the many additions and iterations each modelling method have received. The authors cite ever increasing complexity in CPS and information systems as reasons for employing these techniques. Fault tree and attack tree modelling are very similar to each other, utilising the same top down tree structure with the root of the tree being an undesirable state. The authors identified the primary defence between these two methods as their application context. Fault trees are used to enumerate different scenarios in which the reliability and/or safety of a system is compromised. Attack trees, however, enumerate possible attack vectors to exploit a system.

This review offers an in-depth description of each modelling method as well as many examples of how new behaviour was Incorporated into each. Two well known and implemented improvements were the addition of a dynamic behaviour into fault

trees, enabling the modelling of probability based behaviours, and the creation of Attack-Defence trees (AD trees) which include countermeasures.

Al-Dahasi and Saqib [172] leverage attack trees to provide mitigation suggestions for SCADA systems, specifically in the context of systems related to oil/gas, water and petrochemical industry. Two primary categories of vulnerability were classified - general attacks and DDOS attacks. These categories were then instantiated into two classification trees - the first of which is provided in Figure 5.31.

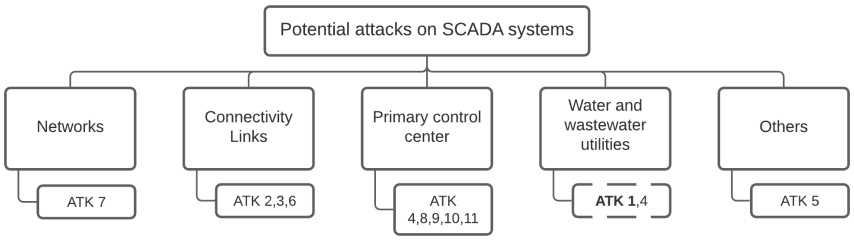


Figure 5.31: SCADA attack tree - potential attacks (adapted from [172])

The authors provide eleven general vulnerabilities, the first of which (ATK 1, Figure 5.31) identifying how physical damage to an IoT sensor can create a cascade effect through deception, DoS or the lowing of system integrity. In addition to the attack tree and the associated DDoS attack tree, they demonstrated how these can be used to identify adversary objectives and aid in countermeasure design.

Buldas et al. [173] describe a method which enables quantitative analysis based off attack trees. The authors identified how current quantitative analysis was unable to handle values of intermediate nodes as well as being unable to represent additional constants obtained from external sources of information, rather than being derived from the tree itself.

To enable this analysis, an attack tree is represented as an *attack tree decoration problem*, which can then be treated as a constraint satisfaction problem - a well known problem in which a solution is an instantiation of each variable that satisfies a set of constraints. To convert the attack tree into an *attack tree decoration problem* a set of Boolean expressions are derived. The names of the Boolean variables are drawn directly from the labels of each element on the tree. These variables are categorised as *Hard predicates* - predicates that the user have no say over - as they constituted the original attack tree. *Soft predicates* can then be defined by the user - these can constitute historical data (such as relevant probabilities) and domain knowledge constraints (such as one event being more likely than another). With these variables defined an analysis engine can classify the problem into three categories:

- Determined
  - There exists one viable instantiation of variables
- Inconsistent
  - There exists no viable instantiation
- Undetermined
  - The problem is neither Determined or Inconsistent

If the problem is not *Determined* the authors provide methods of loosening *Soft predicates* in order to satisfy the problem. This gives the user a method to identify any inconsistency between their historical statistical knowledge and domain knowledge constraints. The authors also provide examples, proofs, and semantic definitions.

Widel et al. [174] reviews literature on attack trees, identifying five areas of research which were popular in either extending attack trees or improving their negative aspects. The primary contribution of this work was the categorisation of research domains and their associated methods. A list of each domain and their associated contributions is given below.

1. Formal Interpretations of Attack Trees
   - Series-parallel Interpretation
   - Linear Logic Interpretation: Specialisation of attack trees
   - Path Interpretation: Correctness of an attack tree with respect to a system
2. Generation Approaches
   - Process Algebra-based generation of attack trees
   - ATSyRA Methodology: Generation of attack trees for physical systems
   - TREsPASS: Generation of attack(-defence) trees for socio-technical systems
   - Biclique Problem for a refinement-aware creation of attack trees
   - Guided Design of Attack Trees by Tracking Useful Positions
3. Static Analysis
   - Pareto Efficient Strategies in AD trees
   - Selection of an Optimal Set of Countermeasures using Integer Linear Programming
   - Efficient Approximation of the Cost of a Cheapest Attack
   - Quantitative Analysis of AD trees with repeated actions
4. Timed Automata-based Analysis
   - Attack Tree Analysis with Priced Timed automata
   - AD tree analysis with timed automata
   - Attack-defence Diagram's Analysis with Stochastic Timed automata
5. Probabilistic Analysis
   - Propagation of Probability Distribution on Attack Trees
   - Combining Bayesian Networks and AD Trees
   - Stochastic Game Interpretation of AD Trees
   - Probabilistic Model Checking for Attack Trees

Formal Interpretations of Attack Trees and *Probabilistic Analysis* align with popular methods used to extend modelling languages to include security.

Kordy et al. [175] provide a method which integrated Bayesian networks into AD trees in order to evaluate probabilistic factors in an attack-defence scenarios, while removing the traditional assumption that all actions involved in the model are independent. The proposed framework is described in Figure 5.32.

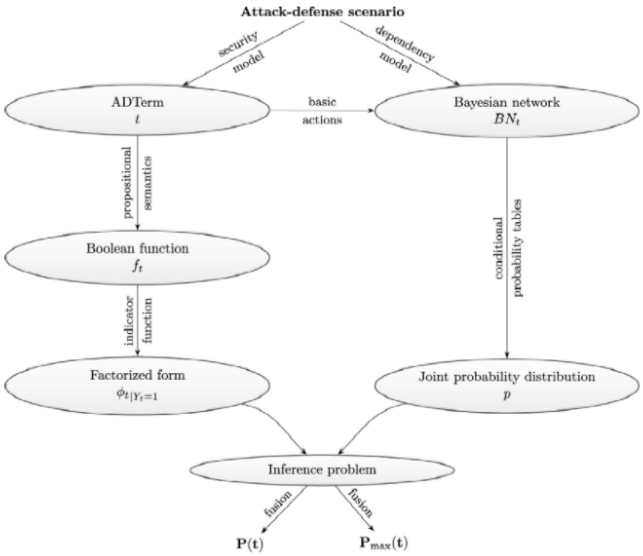


Figure 5.32: Proposed Bayesian/AD Tree Framework (extracted from [175])

As shown in Figure 5.32, a Bayesian network is created alongside the traditional AD tree. From this point conditional probability tables are created which are educated through domain experts and historical statistics. The Bayesian model provides additional probabilistic dependencies between attack steps, enabling attack steps to be dependent on each other - a weakness of traditional AD trees. Finally P(t) (the overall probability the root node - the attack - is successful) and P_max(t) (the success probability of the most probable attack) can be calculated. The authors also offer discussions and methods of increasing the efficiency of probabilistic computations through their provided fusion algorithm and the semiring valuation algebras.

5.3.4.1.2 Attack Graphs

Three primary papers provided insight into attack graphs.

Mao et al. [176] extends the cybersecurity modelling language securiCAD, which is based off the well known Computer Aided Design (CAD) modelling pipeline, with a method of condensing and abstracting models created in the language. The authors identified that securiCAD, and attack graphs in general, often become large and complex - making comprehension of the graphs difficult. To remedy this, they propose a method specifically built for securiCAD of abstraction, reducing the total number of elements and relations shown in the attack graph.

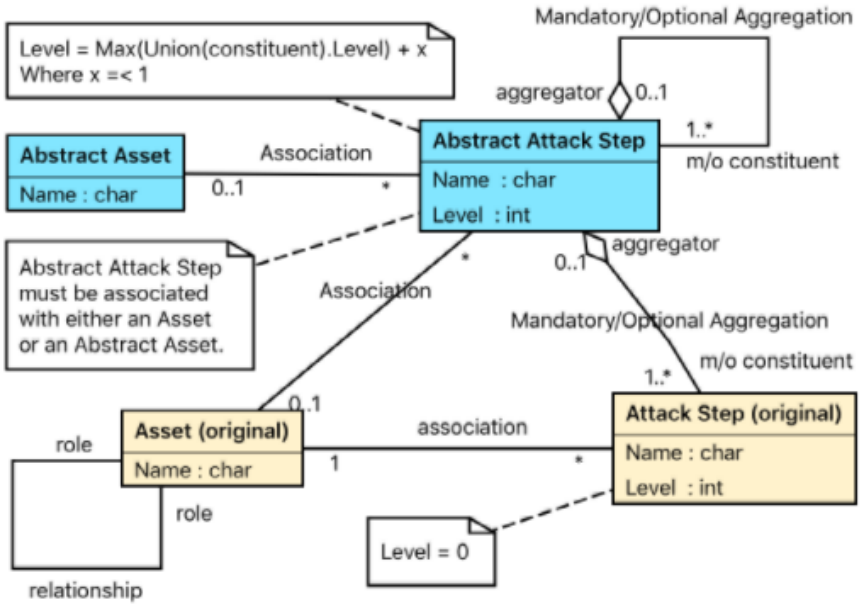


Figure 5.33: securiCAD aggregation extension (extracted from [176])

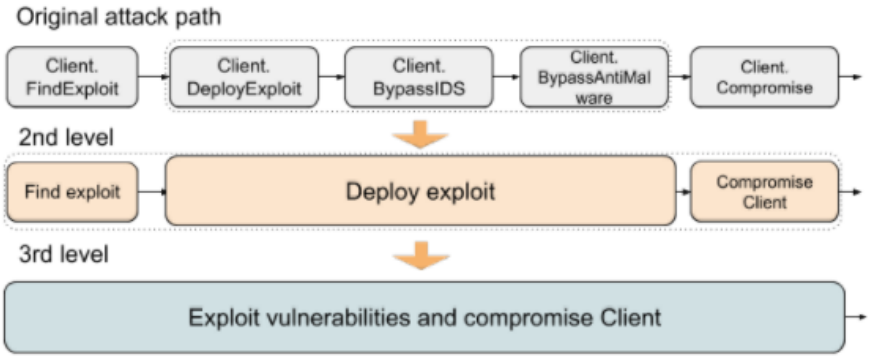


Figure 5.34: Abstracted attack step of Access Control (extracted from [176])

The method relies primarily on two additional elements to securiCAD's meta model (Figure 5.33). *Abstract Attack Step* and *Abstract Asset* enable an aggregation of like attack steps and assets. The abstraction decisions are contextually based on the original securiCAD model and require an expert or analyst to identify like elements and abstract them correctly. An example abstraction is provided in Figure 5.34.

Zhang et al. [177], similarly to [176], discusses attack graph abstraction citing visualisation as a major problem due to the multitude of attack paths in an attack graph. Due to the complexity of attack graphs, and how they are visualised, a distorted risk picture is rendered to both human users and quantities vulnerability assessment models. To remedy this the authors propose a method of abstracting the input data to an attack graph generator, reducing the problem before it has been processed. For example, if their are five identical servers set up on the same network,

these can be abstracted into one entity as any exploit that works on one, will work on all.

Expanding upon this, the authors provide three algorithms designed to identify what elements and relations should be abstracted together. The first: *Reachability-based grouping* identifies hosts with the same network reachability and groups them together. Next, *Vulnerability grouping* identifies similar vulnerabilities on each host and groups them together. Lastly, *Configuration-based breakdown* groups hosts together who are in the same reachability group and have similar configuration (and therefore similar vulnerabilities).

Using these algorithms a complex attack graph can be reduced to vital components for better visualisation and risk estimation.

Khouzani et al. [178] provide a solution to the control selection optimisation problem in attack graphs. The authors provide three primary challenges that effect control selection. First, the effect of security controls is probabilistic and usually mitigate risk rather than remove the threat. Second, controls can affect multiple vulnerabilities and each vulnerability may be affected by multiple controls. Finally, there can be a prohibitively large number of attack paths, making control selection difficult.

The provided solution transforms this problem into an optimisation problem by defining two methods. *The attack problem* is defined as a probabilistic attack graph which describes the *source* of an attack (the initial privilege state of the attacker), its edges (vulnerabilities that an attacker may exploit) and *target* (potential end goals of an attacker). The measure R can then be taken, which describes the attack path of highest probability.

The second method is *the defence problem*. Utilising the previous measure R, the enterprise wishes to minimise this risk by introducing defensive controls. Each control has a set of variables: Cost - the implementation cost of the control, indirect cost - the negative impact on the enterprise that deploys the control, and efficiency - the effectiveness of the control.

Both these problems are transformed into simple linear programming problems and solved for maximising control effectiveness and minimising cost.

5.3.4.2 Topic 2: Risk Assessment (RA)

Four primary papers contributed to RA modelling.

Sommestad et al. [179] devises a new method of analysing security risk in a system architecture by utilising PRM's and their ability to include probability metrics. The authors identify how, while decision makers usually have a reasonable understanding of the system architecture, their understanding of the interactions between security measures, threat environments and sensitive assets is ambiguous. To help remedy

this, they first provide a method of instantiating PRM's into system architecture, and then provide a package of abstract PRM classes which can be used to infer security risk from architectural models.

To include PRM's into an architectural model the authors provide a set of methods describing how causal relations and abstract/concrete PRM packages can be defined - an example is provided in Figure 5.35.

The attribute *Availability* in Figure 5.35 provides an example of how a PRM is able to infer a probability value for the availability of the system through aggregating probabilities from the associated attributes *Reliability* and *Competence*.

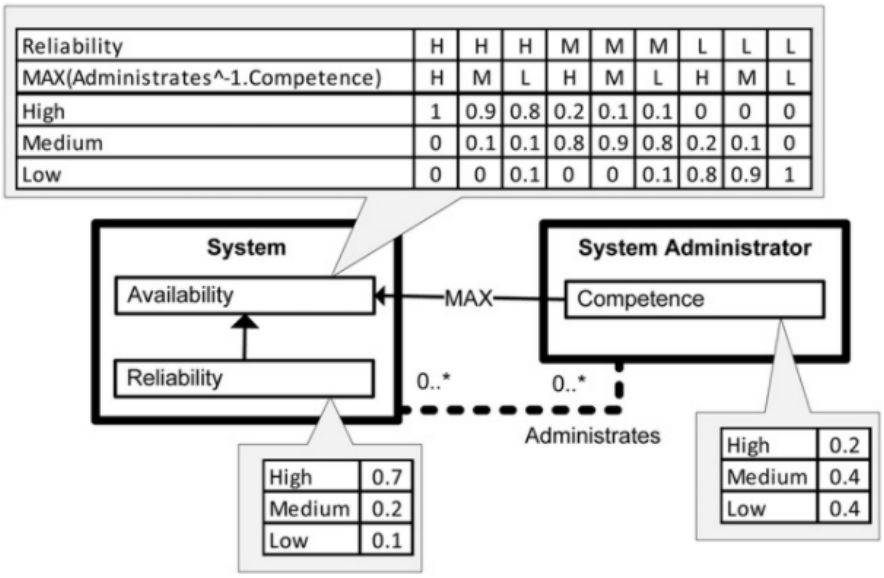


Figure 5.35: Example PRM meta-model (extracted from [179])

Lamine et al. [180] discuss risk management in the context of enterprise engineering and architecture. With the growing interest of risk in enterprise engineering, the domain of risk-aware business process management (R-BPM) has become popular in the academic domain. The authors provide their own contribution in the form of a new management framework and modelling language, Business Process Risk Integrated Method (BPRIM), which complements both business process management (BPM) and risk management.

First, BPM and Enterprise Risk Management (ERM) lifecycles are integrated, creating a joint life cycle (Figure 5.36) consisting of four phases: *Contextualise*, *Assess*, *Treat*, and *Monitor*.

Next, a meta-model was defined for BPRIM based off the devised life cycle containing the primary concepts handled during each stage and their associated relationships. Finally a modelling language was devised to support BPRIM's elements and relations, drawing upon the Extended Event-Driven Process Chain (eEPC) modelling language. eEPC was decided upon as it already incorporated the majority of

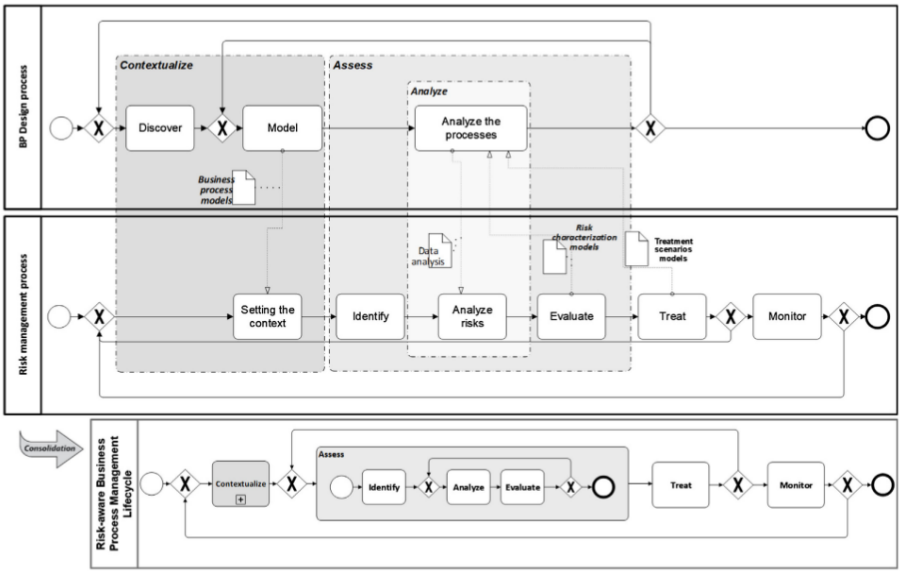


Figure 5.36: Joint BPM and ERM lifecycles (extracted from [180])

the required concepts while also supporting a view based approach. The authors also designed and implemented tooling to support this modelling language, proving a means for their approach to be utilised.

Latvala et al. [181] develop a risk visualisation technology to describe and map risk elements and their associated relations. The authors identify that risk visualisation can be difficult due to the huge number of risks with diverse causalities. To remedy this a tool is designed which enables the user to expand upon a certain element via clicking on the element in the interface. They are then presented with the associated risks, security objectives and mitigating controls and how they interact. In this multiple levels of complexity are able to be condensed into a tree like structure.

Hall et al. [21] provide a discussion on risk visualisation from the perspective of a wider information visualisation framework. The authors introduce three categories of visualisation, *Journalistic*, *Scientific* and *Critical* visualisations. Risk visualisation falls into the last of these categories, *Critical*, enabling user-lead enquiries into technical and cultural risks of a system. The authors provide a case study, in which they enable critical risk visualisation through a series of Lego scenarios, allowing users to work collaboratively in identifying cultural and safety risk scenarios.

5.3.4.3 Topic 3: Architectural Automation

Three primary papers contributed to architectural automation.

Buschle et al. [182] develops a method of automating the creation of EA models through utilising a network scanner, EA analysis tool and security modelling lan-

guage. The authors provide a solution to the complexity of EA models. The creation of EA, organisation-wide models often culminates in cumbersome models due to the nature of including multiple aspects of an organisation into one model. To combat this the authors provide a method of automating the process through a pipeline with two components.

The first component, the vulnerability scanner NeXpose, provides information on the networks architecture, identifying all devices on the network which are communicating over either the TCP or UDP protocols. Credentials are able to be given to NeXpose in order to map protected environments to produce a complete architecture of the network.

Next, a previously developed EA analysis tool [183] is used to both create a meta-model of the architecture and create the actual instantiation of the architecture which is compliant with the previous meta-model. The authors extended this previous work with an extension designed to take advantage of the NeXpose scans, using these scans to automatically instantiate the meta-model.

The authors offer another advantage born from NeXpose, showing how the NeXpose output can also automatically generate a set of entities, relationships and attributes for the modelling language CySeMoL. While not producing a full model, the generation of these select items reduces the effort required to generate the entire model.

Lagerström et al. [184] explores the idea of automating architectural "to be" models by using value heuristics and learning algorithms. While other works have provided methods of automating the creation and analysis of architectural models, methods that automate the improvement of architectural models are scarce. Experts in the field are required to perform manual improvements on the architecture, offering insight into cost effective and efficient solutions for the Enterprise.

To automate this process the authors provide an algorithm that proposes changes to an existing architectural model. Based on the Markov Decision Process the algorithm is given a finite set of actions it can perform on the model, and a heuristic to guide the algorithm on which action to perform. The heuristic chosen to educate the algorithm is a multi-factor joint utility function (example in Figure 5.37) which can replicate an expert's understanding of the domain area. For example, the utility function can take factors such as *time to compromise*, *cost of implementation* and *availability of the system* into account.

Finally, the reinforcement learning algorithm *State-Action-Reward-State-Action* (SARSA) is used to identify the security policy that outperforms all other possible policies, maximising the total reward based off the above heuristic.

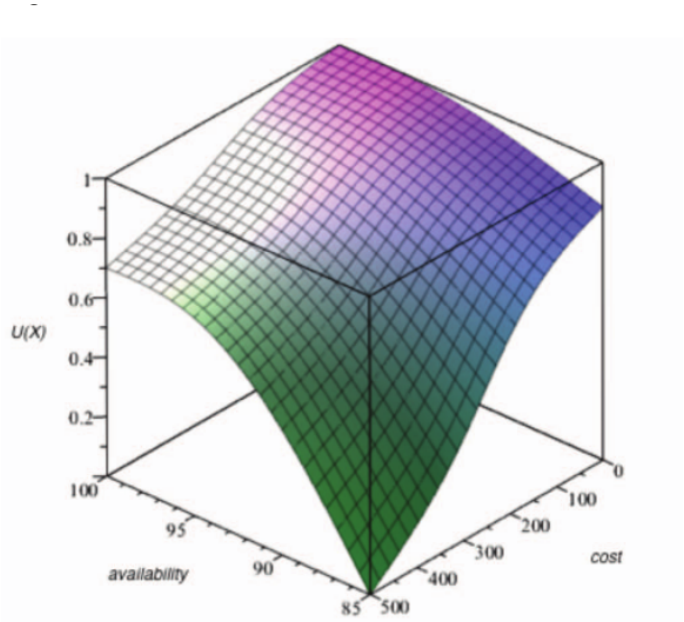


Figure 5.37: Example joint utility heuristic (extracted from [184])

Falessi et al. [185] builds on their previous work, [186], and develops a tool to automatically extract safety-relevant elements of a systems architecture to aid safety assessors in inspections. The authors highlight the difficulty faced by safety assessors as they are often required to browse through extensive system models in order to identify safety-relevant aspects to ensure they are up to standard. To reduce this overhead, SafeSlice was designed which automatically highlights and extracts safety relevant elements ready for inspection. This is achieved primarily through strong traceability enabled by both SysML (a de-facto system modelling language) and, the authors own traceability extension to SysML discussed in [186]. Through traceability, safety requirements are linked to their associated system elements. An element of SafeSlice, *Rule Assistant*, is also enabled through traceability, providing the ability to apply rules to safety elements thus, guiding the user to conform with safety standards.

Once traceability is established, slices are able to be generated which extract related elements from the system in regards to safety requirements. These can then be inspected through a provided tool, *inspection Assistant*, enabling the quick inspection of all safety related elements. An overview of SafeSlice's architecture is provided in Figure 5.38.

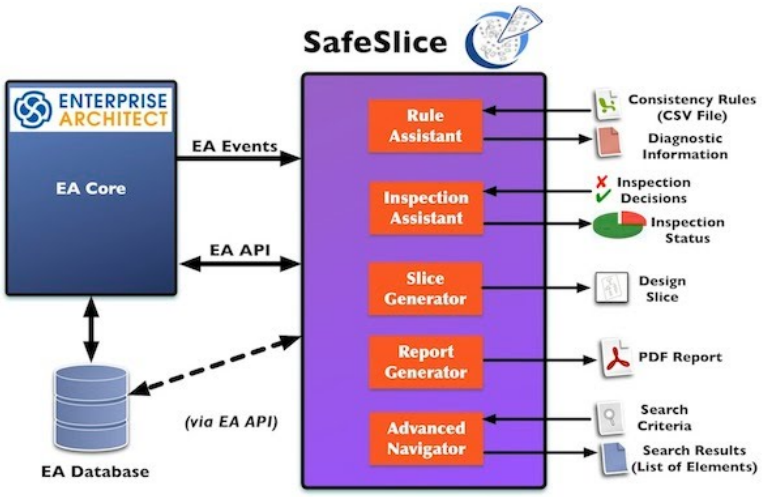


Figure 5.38: SafeSlice architecture (extracted from [185])

5.3.4.4 Topic 4: Cybersecurity Modelling Languages

Five primary papers contributed cybersecurity modelling languages.

Easttom [187] defines a set of new security diagrams and elements, based of SysML and UML, for application in the cybersecurity domain. The authors identified how, as cybersecurity is a relatively younger domain than computer and software engineering, there are less modelling tools afforded to it. To design a new modelling paradigm the authors took inspiration from the SysML and UML modelling language, using diagrams described in these standards to educate their proposed modelling language. In total, four new/modified diagrams for application in cybersecurity were proposed: the *Misuse Diagram*, *Security Sequence Diagram*, *Data Interface Diagram*, and the *Security Block Diagram*.

The *Misuse Diagram* extends SysML and UML's case diagrams, providing four new contextual elements enabling the modelling of misuse cases. The *Security Sequence Diagram* contributes a slight adjustment of the traditional sequence diagram found in SysML, providing one new contextual element which describes an unauthorised sequence. Next, the *Data Interface Diagram* is a completely new addition to this language. It was designed to provide information on incoming and outgoing data flows in order to describe security vulnerabilities during data transit. Finally, the *Security Block Diagram* was based off UML's component diagram. The authors modified this diagram with three new block categories designed to enable a data flow oriented diagram, assisting analysis's with data flow from component to component.

Ekstedt et al. [188] present a CAD tool, securiCAD, designed for application in the cybersecurity domain. The original goal of this application was to provide tools, similar to CAD tools used by traditional engineers, for IT professionals in order to aid in the design and testing of networks and EA's. SecuriCAD provides four value

propositions - *Access to expertise*, *Improved cyber security management efficiency*, *Improved visualisation*, and *Improved cyber security ROI*. To enable these securiCAD provides a framework of asset types (protocols, web applications, operating system, etc.) and associates each asset with its applicable cyber attacks, vulnerabilities and security controls. The probabilities associated with an attack are elicited from experts and scientific studies which further enable RA, providing the analyst with information on the most vulnerable elements in the architecture.

Sommestad et al. [189] develops the cybersecurity modelling language CySeMoL and provides a validation of the underlying PRM used in its operation. CySeMoL utilises a PRM structure defined in previous research [179] which outlines a meta-model for a cybersecurity based PRM. This meta-model can enable two types of analysis. the first type of analysis provides expected economic losses due to a successful event and requires all conditional probabilities of the PRM to be defined. The second analysis provides a probability indication regarding the likelihood of a successful attack. CySeMoL enables the second analysis through identifying a subset of classes, attributes, and dependencies identified in [179].

The authors offer a set of classes with their associated reference slots and attributes educated through a literature study and domain expert reviews. These provide the qualitative structure of the new PRM, identifying what probabilities need to be elicited before analysis can begin. Further defining of the PRM is performed through identifying what quantitative parameters will need to be elicited. This is done through two methods - identifying *Logical Deterministic Dependence's* and identifying *Probabilistic Uncertain Dependence's*. Once done, the defined probability variables can be assigned through expert judgement (using Cooke's classical method), scientific experimentation and data collected through previous work. In total the developed PRM and CySeMoL contained 22 classes, 102 attributes and 32 class relationships (reference slots).

Holm et al. [190] extend CySeMoL, discussed in [189], in order to remedy limitations relating to the scope and implementation of the modelling language. The authors provide three primary motivations and advantages for their new extension, P^{2}CySeMoL. The first motivation is that CySeMoL was originally introduced in the context of SCADA cyber security modelling. Because of this, an emphasis was placed on security attributes critical to SCADA systems, for example, availability and integrity. P^{2}CySeMoL remedies this by including a wider scope of elements and relations, for example the addition of the element *NetworkVulnerabilityScanner* and *WebApplication*. These additional elements widen the application scope of the language and provide more contexts in which it can be utilised. The second motivation was that during attack path calculations, CySeMoL assumed a work week worth of time for the attacker to compromise the system. P^{2}CySeMoL has customisable work

hours per attacker which can be prescribed by the user, enabling contexts in which prolonged attacks may be realistic. The last motivation was due to CySeMoL's poor computational cost when computing larger models. This limitation was due to the PRM which was switched out for a *Predictive, Probabilistic Architecture Modelling Framework* (P^2AMF) which allows the attack step domain to be pruned by excluding all non-reachable attack steps. While this method leads to a larger computational cost on very small models, it provides a massive decrease in computation cost in larger models.

Demir et al. [45] builds upon their previous work, [191], adding functionality to abstract cross cutting features, such as security, from the architecture. The authors identified that, when security features are integrated into distributed systems, they are often written into each module individually - even through they constitute the same security control. To abstract these security features into their own model-driven compliant module, the authors provide a theoretical new cross-cutting module that can facilitate these cross-cutting features. The new tool, DISCOA, implements a module that can be inserted into an architecture between communication paths of other modules, allowing an action or check to be executed before processing continues. For example, a client needs to provide payment information to a server however there are no security checks on what the client is sending the server. Using DISCOA, an architectural adaptation can be designed and bound between these two roles which provides the required security.

This method allows for reuse - the solution can be bound between different sets of roles, providing the security abstraction that the authors sought. The solution also benefits model-based engineering, providing a separation of concerns and the ability to specify security features as a module in the architectural description.

5.3.4.5 Topic 5: Other

5.3.4.5.1 Literature Reviews

Zhou et al. [192] provide an SLR created to reduce the gap of knowledge between the purpose and means of EA visualisation methods. The authors identify EA visualisation as a primary enabler of EA, and developed a study to identify its current state in literature. Six RQs were defined, three of which are of primary relevance - RQ1, RQ2 and RQ5.

RQ1 identified the main purposes and motivations of EA visualisation. The authors identified six primary purposes and motivations for EA visualisation in academia. The most relevant of these motivations is security analysis, which the authors identified as being a dominant theme in the literature. Much of the literature follows similar themes to reviewed literature in 5.3.4, using Bayesian statistics and graph theory to enable risk analysis.

RQ2 identified trends in the historical development in EA visualisation. Two trends were identified. First, a large growth of publications regarding EA visualisation was found past 2009 which the authors attributed to two possible causes - the publication of ArchiMate 1.0, and the growing need for governance in large enterprises. The second trend where research dropped off substantially during 2014 and 2016, the authors were unable to identify any possible cause for this.

Finally, RQ5 identified techniques that are used in the study of EA visualisation, and their pros and cons. The authors offer comparisons of both visualisation languages (ArchiMate, UML) and associated EA frameworks (TOGAF, DoDAF) using criteria based on the 5W1H interrogatives. ArchiMate, in general, was found to be the most expressive visualisation language.

The last contribution of this paper is a meta-process of EA visualisation, educated from the SLR content. The authors identified six phases that define the majority of visualisation approaches (Figure 5.39).

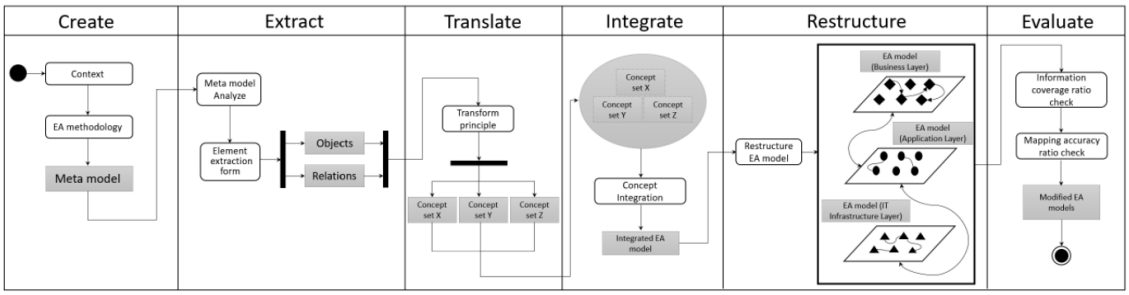


Figure 5.39: Proposed EA visualisation approach (extracted from [192])

Finally, an important analysis to highlight in this work is the dominance of ArchiMate in the EA visualisation domain. ArchiMate was utilised in 41 of the found primary papers with UML being the second most utilised at 13 papers.

Mažeika and Butleris [193] provide a review of security requirements engineering processes and modelling methods in order to educate a new MBSE security profile. The motivation for this work stems from the lack of representation security aspects are afforded by both the SysML language or MBSE methodologies. By integrating security aspects more directly into MBSE, the advantages afforded to MBSE (reduced risk, managed complexity, reuse) can be applied to the security domain. To do so, the authors propose a review process comprised of four elements. First, an analysis of related work, followed by a conceptual alignment between modelling approaches, creation of a security domain model, and finally, the creation of a security profile for MBSE.

During the literature review the authors identified four modelling approaches, the UAF, the Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS), SysML Sec, and UML Sec. These four approaches are com-

pared through a conceptual alignment, identifying which, out of eight concepts, each approach supports.

A security domain model is presented, consisting of three elements - *security assurance concepts* (concepts that ensure system security or mitigate risk), *items to be protected* (vulnerable assets in the system), and *risk-related concepts* (risks and system weaknesses).

Finally, a security profile/framework is presented basing its propositions on the ISO/IEC 27001 standards, and the previously created security domain model. The profile consists of five steps - define the RA approach, identify the risks, analyse and evaluate the risks, identify and evaluate options for risk treatment, and finally, select control objectives and risk controls.

Xiong and Lagerström [194] discusses a SLR in the domain of threat modelling in order to answer the two RQs, "What is threat modelling?", and "What is the state-of-the-art work in this field?". This work was motivated through an observation that threat modelling was widely applied in many contexts, providing many different interpretations of what threat modelling indicated. To produce a unified understanding of threat modelling, the authors performed a SLR which identified three clusters of research - C1: The application of threat modelling, C2: Threat modelling methods, and, C3: Threat modelling processes. C1 and C2 constituted the majority of the selected primary papers, with C3 only making up five of the fifty four primary articles.

The authors identified that, generally, threat modelling "is a process that can be used to analyse potential attacks or threats, and can also be supported by threat libraries or attack taxonomies."

Overall, the authors found that the future research on threat modelling was revolving around automation, validation, inclusion of defensive measures, and expansion of threat categories in specific application domains.

5.3.4.5.2 Miscellaneous

Santos et al. [195] develop a functional system architecture for Air Traffic Management (ATM) designed to provide support for safety related functions. The authors cite an increasingly complex, automated and tightly coupled ATM systems increasing the difficulty of providing safety management as their motivations for introducing the *Model of ATM Reality in Action* (MARIA). To handle the increasing complexity of safety management, MARIA aims to provide an architectural basis for system analysis, primarily for safety, by describing ATM system components and their relationships.

The authors deliberated over three possible modelling methods to utilise in their research - BPMN, Functional Resonance Analysis Method (FRAM), and, Structured

Analysis and Design Technique (SADT). BPMN was discarded due to three primary reasons - its relative complexity requires training, the inability to model stochastic events, and, the functionality of BPMN being based around a start event and end event. FRAM was discarded primarily due to its exponential complexity found in bigger models, and, ambiguity regarding elements and their contextual meaning. SADT was the chosen modelling methodology for the solution.

To populate the model with elements and relations the authors used expert knowledge, a literature review and exploratory interviews. Once developed, the model was verified internally and externally - offing the model for evaluation to enterprises who had not part in its development. In total, nine abstract functions were identified in the ATM domain (Figure 5.40).

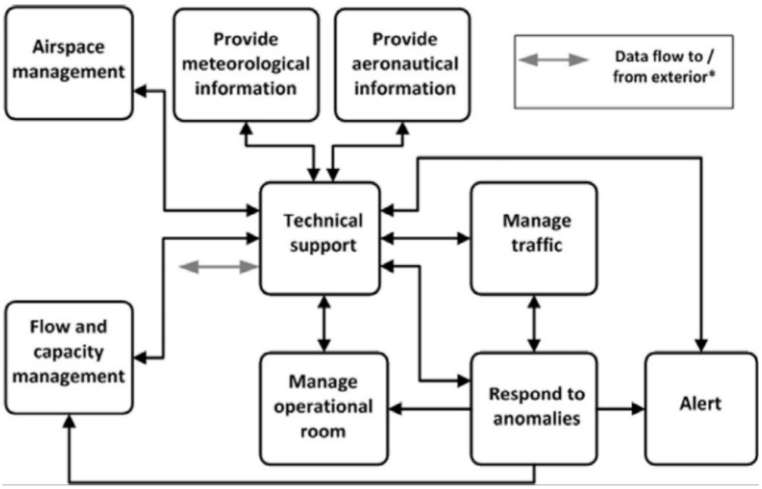


Figure 5.40: Top level ATM functions (extracted from [195])

Buschle et al. [183] design a tool to integrate PRM functionality into EA models to support analysis and decision making. The introduction of a PRM into EA enables an array of analysis options dependent on the context the model was created in. For example, if an EA model had the goal of providing improvements in the cybersecurity domain, elements relating to this context would be included in the model which can then be instantiated into a PRM which allows analysis of the cybersecurity in that model. In this way, many different types of analysis are enabled, dependent on the structure of the model.

The authors define EA analysis as a three phase process consisting of the *Assessment Scoping*, *Evidence Collection*, and, *Analysis* Phases (Figure 5.41). *Assessment Scoping* refers to goals the architect wishes to represent in their model, denoting their understanding of the domain and the relationships between elements and their attributes. *Evidence Collection* produces quantitative information to facilitate their analysis. Evidence collected is often not infallible, and as such a credibility rating can be applied to each instantiated attribute, allowing these uncertainties to be rep-

resented in the model. Finally the PRM can be used to provide comparative analysis on the current model and a next possible iteration.

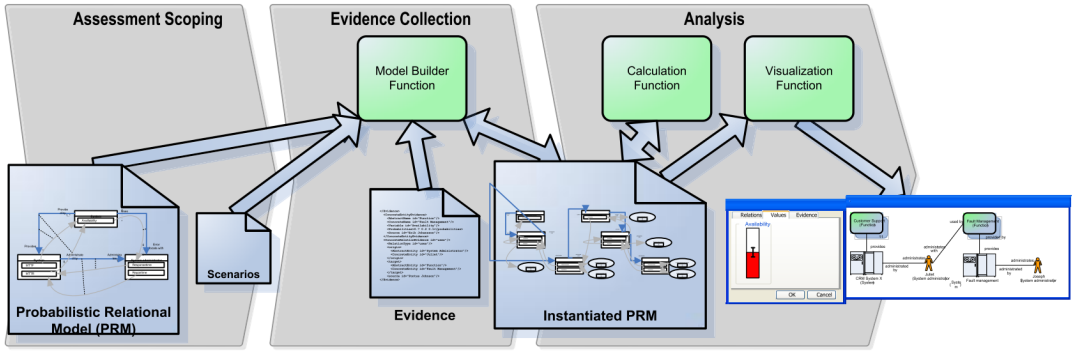


Figure 5.41: The EA analysis process (extracted from [183])

Chung et al. [196] utilises architecture descriptions to provide a platform for penetration testing. The authors present a pen-testing methodology which reconstructs a software's architecture in order to enable efficient decision making by the pen tester. The provided methodology includes a methodology for re-documentation described in [197], which describes how to backwards engineer software into a documented architecture consisting of 4+1 views [198]. These views are then used to discover abuse cases within the architecture, which the pen tester may try to exploit.

Chapter 6

Discussion

This chapter provides a discussion on the research questions through synthesising information found during the SLR combined with interview findings through the method of externalisation. First, each RQ is enumerated upon, discussing findings and meaning drawn from its associated primary papers. Next, insights are extracted from the interviews. Finally a set of observations are made regarding the research domain and future research opportunities are identified.

6.1 RQ1

How are security aspects being incorporated into ArchiMate EA Models?

Three general research trends were identified from the six primary papers who constitute this research question: ArchiMate's coverage of certain security applications, the incorporation of new behaviours in ArchiMate and finally, agile processes utilising ArchiMate.

6.1.1 ArchiMate's coverage of security applications

Three primary papers utilise ArchiMate to enable threat analysis [165], security assurance cases [169] and, risk assessment modelling [35].

Two of these papers apply ArchiMate in the duel contexts of EA and security modelling [165], [169]. Berkel et al. [165] utilise ArchiMate to provide a framework architecture for smart cities which includes a cross cutting column in the architecture - security (Figure D.1). Implementations of security or other viewpoints with opposing architectural drivers often constitute a cross cutting viewpoint or column in the framework. Various issues arise with this method which is further explored in section 6.6.6.

Zhi et al. [169] provides a visual assurance method utilising ArchiMate in the context of EA. This solution provides assurance cases and goals to be integrated

into an EA, giving assurance arguments between enterprise elements. The method used to achieve this, similarly to [165], results in a cross cutting architectural driver (Figure D.2).

These two papers were both published in the second wave of research on this topic (2018-2020, see Figure 5.3) after the inclusion of the RSO by TOG in ArchiMate 3.0. Korman et al. [35] identified ArchiMate's capability in regards to modelling risk assessment methods. As this was published in 2014, the RSO was unavailable, and as such, the authors developed their own alignments between ArchiMate elements and the required risk constructs. This research, however, did not include the context of EA, which could be due to the emerging nature of the joint ArchiMate, Security and EA topics.

A fundamental issue with these approaches is the divergence from the quality attribute *compartmentalisation* which is achieved through the traditional EA layer approach. *Compartmentalisation* plays a large role in complexity management – a fundamental design consideration for multi-view modelling approaches, and by including a cross-cutting layer which intersects with multiple layers, this quality is lost. This outlines one primary future work opportunity – identifying how cross cutting elements can be included within a multi-layer modelling method without sacrificing the benefits.

ArchiMate 3.0's full architecture specifies the inclusion of the aspect, "Motivation", which is instantiated in the framework as a cross-cutting element. Comparing the motivation aspect with the security layers proposed in [165] and [169] provides context on how TOG engineered cross cutting elements into the framework while maintaining *compartmentalisation*. Figure D.3 describes the meta-architecture of the motivation aspect and shows that each motivational element is associated to one stakeholder, who is associated to a structure element. In this way, cross cutting relationships between core elements are avoided as each core element would instantiate its own motivation aspect; i.e., motivation aspects are not shared across core elements. This method, while useful in the context of motivation elements, is inoperable in regards to security elements as security often spans many affected core components.

6.1.2 Incorporating new behaviours into ArchiMate

Hacks et al., [166] and Närman et al. [167] provide extensions to the ArchiMate language in order to enable quantifiable security evaluations of different systems. Hacks et al. [166] provides a translation method, converting ArchiMate models into MAL models in order to simulate attacks on the chosen system. Närman et al. [167]

proposes a method for availability analysis, integrating a PRM structure into ArchiMate in order to provide a statistical quantitative model. Statistical measurement, and PRM implementations especially, are often presented as options for security based model analysis and are discussed in more detail in section 6.6.3.

Papers which extend the ArchiMate language follow a trend, often aiming to provide a form of quantifiable evaluation to the modelling language. This is due to the lack of these measurement systems in the modelling language itself, which is an opportunity for the enrichment of the language. However, there may be a disconnect between the objectives of these applications. *Formal Modelling*, in this context, identifies models in which you seek quantifiable information, such as statistical models. ArchiMate, while being able to support these systems, may not benefit from these inclusions as it is contextually utilised as a tool to present a visual diagram of an enterprise. The enrichment of these evaluation methods may prove to be both cumbersome and unnecessary in this context.

Formal Modelling methods are useful in many contexts. However, integrating them into ArchiMate is inefficient as the application domain, and thus the design of ArchiMate, is specifically for communication and visual modelling. Incorporating formal modelling into ArchiMate through a different mechanism may be useful. For example, a core element could describe a statistical model which is used to generate a performance metric. Including this model as an element in the architecture acknowledges its utility within the enterprise, and can potentially represent complex decision processes such as risk posture.

6.1.3 Agile security processes using ArchiMate

While only one paper discusses using ArchiMate in an agile context, it is increasingly important to understand the limitations of EA, and the benefits of re-usability and modularity. EA documentation, which ArchiMate models are classified as, requires a large amount of investigative work and investment by an enterprise architect in order to implement iterative improvements. With the rise of quickly evolving technologies, enterprises have the ability to evolve their IT and technological systems at a fast rate, increasing the overheads required to ensure that the EA model and its associated documentation is kept up-to-date.

Xiong et al. [170] proposes an agile threat modelling method which utilises ArchiMate. The agility of this method is based on an EA-repository, which stores precomputed information, enabling the analysis of components and architectures. The primary limitation of this method is the reliance on the repository, which would require significant on going work and input data in order to be functional.

ArchiMate enables agility through its layered approach, providing compartmentalisation between components and traceability through relationships between elements. This allows an architect to identify affected components during a model iteration. Depending on the abstraction level of the model (the granularity of the processes being recorded) the effort to manually identify changes to the model could increase substantially. One proposed solution to this problem could be architectural automation as discussed in Section 6.6.4.

Agility, while required in EA modelling, is still far from being achieved given the complexity of modelling context and available tooling. Complexity is a primary concern in EA, and is one of the founding reasons for implementing EA modelling. Future research regarding the handling of complexity in this domain would be valuable.

6.1.4 Conclusions

In conclusion, there are three primary methods being used to include security aspects into ArchiMate EA models. Two of these methods utilise ArchiMate concepts and are delineated by their approach to the EA context. One of these ignores the overarching application domain of ArchiMate ([35]), and the other attempts to integrate security modelling into the EA application domain ([165], [169]). The third method is the development of new security features within ArchiMate as discussed within [166] and [167].

The second wave of ArchiMate research (Figure 5.3) predominantly presents EA and security as joint topics, while earlier work focused on security being isolated, or separate. This outlines the movement within academia to research these topics together, as there is more recognition of the importance of security modelling in EA.

The current methods of security implementation in ArchiMate may be at odds with its primary application. ArchiMate models serve as forms of visual EA documentation, and as such practitioners are less likely to identify them as *formal modelling* opportunities. Visual additions to ArchiMate to support security modelling often result in cross-cutting architectural drivers which reduces the compartmentalisation of ArchiMate's layered approach. Implementing Security as a cross cutting element is an increasingly researched topic in the EA application domain, supporting a growing interest in the governance of these systems by relevant stakeholders.

6.2 RQ2

What elements are required in ArchiMate to model security aspects in the context of micromobility?

To answer this research question, four different contributions were identified. First, potential attack vectors such as physical attacks are discussed. Second, relevant quality attributes of security in the micromobility context are provided. Third, relevant elements, instantiated in parallel domains are presented and Finally, overall design considerations regarding the modelling of micromobility are provided.

6.2.1 Understanding attack vectors

Classification of attack vectors extracted from relevant primary papers follows the *Social*, *Physical* and *Cyber* convention used in [66]. Distinguishing attack vectors based on these categories is useful in the context of micromobility technology as physical attacks on these systems are much more probable than other instantiations of IoT/CPS technologies. Vehicles offered by the micromobility services are provided in public, under little supervision, which increases the risk of tampering, theft and general damage.

Table 5.5 presents extracted physical attacks from relevant primary papers. Physical attacks in the AV, IoT and CPS domains present a general theme of the damage or modification of hardware. The risk of direct physical damage to a micromobility vehicle, either through malicious acts or general wear and tear, is less mitigated by security considerations and aligns closer with the general rigidity, engineering, and social perception of the micromobility enterprise. Physical access to USB ports - or any interface with software - provides a more security-centric problem.

An interesting inclusion to the physical attack domain is resource depletion attacks. These attacks have been identified in IoT and CPS technologies as a method to disable security countermeasures and access an intended function. In the context of micromobility vehicles, this attack can be used to disable a vehicles GPS and/or backup GPS locator, providing the opportunity for modification and theft before the enterprise can dispatch their retrieval staff.

Cyber attacks constitute the majority of extracted attack vectors (Table 5.6) and identifies many attacks including networking, code injection and access control. DoS and DDoS attacks have been, and still are a major issue for services who rely on server bound services.

The risks of DDos and DOS attacks become increasingly clear with micromobility enterprises who rely on the connectivity between mobile apps and servers in order to unlock the vehicles, and provide their transport service. If the servers are unable to service clients requests, the provided vehicles become unusable - an issue which becomes increasingly problematic as a micromobility enterprise scales up. If micromobility is to be heralded as a new method of urban transportation, the meth-

ods used to implement it should not rely on servers (or should have an alternative method of unlocking a vehicle) as it would perform a critical transport function.

Other cyber attacks critical to the micromobility domain include identity faking, replay, and impersonation attacks in which an attacker attempts to gain access to a vehicle through exploiting a weakness in the implemented authentication mechanics. As micromobility enterprises offer a direct and physical service, these attacks become desirable as a means to access this service for free.

Micromobility vehicles, similarly to connected vehicles and AVs, often have wireless connectivity - either to connect to a server or to connect via Bluetooth to a personal device. Bluetooth, specifically BLE (Bluetooth Low Energy) - a form of Bluetooth used in resource constrained devices - has been demonstrated to be vulnerable to many different types of cyber attacks. As such it would be recommended that this connectivity be either excluded from micromobility devices, or, implemented in such a way as to not provide access to critical systems of the vehicle. Aside from these primary cyber attack concerns, there are also many attack vectors related to networking, input validation, and access control which need to be accounted for.

Social attacks constitute the smallest set of attack vectors and identify attack vectors which originate from employees and potentially social actors such as protesters, white hat hackers, and others. Examples provided by primary papers include the privileged insider attack in which an employee who has access to internal systems compromises an area of the enterprise. Another attack includes social engineering, in which an employee is tricked into disclosing sensitive information or credentials through phishing attacks. These security vulnerabilities are treated with personal training, vetting, and general awareness of these risks.

In general, attack vectors were discussed often without an associated mitigation technique. This could be due to how strongly each solution is context bound. Another issue is the fact that many micromobility enterprises utilise third party software, services, and technology to enable certain portions of their functionality. It is difficult to understand what security concerns could arise from the inclusion of these service, and how - if multiple third party solutions are used - the combinatory nature of these third party solutions affects security. This provides a primary question for future research - “how should third party services, and their potential security considerations, be modelled in an EA?” which was also a theme discovered during industry interviews (Section 6.5.1).

6.2.2 Security Quality Attributes and Application Domains

Security quality attributes provide direction and frameworks for areas of security development in applications. Identifying what security attributes are used in differ-

ent domains can help identify the differences between application domains (such as IoT, CPS, AVs, etc.) and helps classify the goals of each security solution.

In general, all primary papers which reference security quality attributes include the classical CIA triad. More specific security attributes were identified in different application domains. In the IoT domain, Omoniwa et al. [69] identified *Trust*, *Authentication*, *Privacy* and, *Access Control*, outlining the importance of information integrity and confidentiality in the context of generic IoT systems. Kavallieratos et al. [74] identified *Authenticity*, *Possession and Control*, *Utility* and, *Non-Repudiation* in the context of the CPS application domain, providing a similar emphasis on access to information and confidentiality.

The similarities between the security attributes of these domains (IoT and CPS) is understandable as they are similar in their instantiations. Baloyi and Kotzé [46] observed that CPS frameworks often constitute a meta-framework for IoT instantiations, making it possible to represent other IoT and CPS architectures using CPS meta-architectures. This closeness in application, architecture and, technology may explain why they have similar security quality attributes.

These specific security attributes constitute two of the three pillars of the CIA Triad, Confidentiality and Integrity. The micromobility context affords a unique application of IoT/CPS in which the final pillar, Availability becomes more important. For reasons stated in section 6.2.1, availability of critical services such as transport should be emphasised in these applications.

An emerging theme in the SLR is the joint consideration of both Security and Safety, which are provided in tandem [74]. Like the CIA Triad, Security and Safety have many overlapping applications, as well as potentially opposing applications. For example, if a user pre-pays for one hour of ride time on an electric scooter, the scooter should not force the use to stop after this hour is up for safety reasons. This could also be treated as an exploit and therefore, as a security concern. The joint elicitation of Security and Safety identifies these conflicts and seeks to provide an adequate countermeasure. Joint quality attributes such as availability are also possible. Availability in the context of security guarantees the service is accessible by authorised users, however, in the context of safety, identifies that the device should be able to provide a stated function if demanded under given conditions over the devices lifetime.

The importance of safety is another unique aspect to micromobility services as, by offering vehicles for public use, the risk of bodily harm is greater than traditional instantiations of IoT and CPS technology.

Two papers identified applications of CPS and IoT technologies with both identifying Intelligent Transportation – which micromobility is an instantiation of – as a growing application domain. Many of these applications intuitively include privacy

as a primary security topic (smart home, smart health-care, smart cities, etc.) which highlights a growing movement towards privacy regulation and governance. This is discussed in more detail in section 6.6.2.

6.2.3 Instantiated Elements and Vulnerable Components

Identifying and extracting already instantiated modelling elements designed in the joint security - IoT/CPS/AV domains indicates what elements researchers believe hold the most utility to their application. This is useful as it identifies common elements, components, and relationships as well as their associated security considerations.

Tables 5.3.1.2 and 5.3.1.2 outline common and unique elements found for application in the vehicle modelling and IoT domains respectively. Security modelling of vehicles and vehicular systems was a relatively popular topic with five primary papers providing modelling elements.

Two primary papers, [75] and [62] provided the majority of the common elements with both identifying an ECU and four types of networks. A general network element, *VehicleNetwork* is included as well as two vehicle network connectivity standards - the *CAN* and *FlexRay* networks. An entertainment and sound system network, the *LIN* network is also included however is not applicable to micromobility vehicles as these do not include these technologies.

Unique elements span from physical technologies such as GPS [70], to generalised concepts such as Dataflow [62]. This is due to their application context as these elements are designed to work within the overarching context of the articles they are used within. These unique elemets generally enabled one of two mechanisms. First, the creation or extension of a formal modelling technique in order to run simulations or provide quantifiable security metrics. Second, to provide a graphical representation of the security of a system to a user. For example, [73] and [79] both introduce visual accents into their element design in order to promote understandably and readability of their respective models.

The identification of vulnerable components provides a basis of which elements in a technology need to be included in a security model, in order to mitigate these vulnerabilities. The extracted components can be found in Table 5.3.1.1.2. These components create a solid basis in the development of elements for security modelling of micromobility applications. A list of micromobility-compatible components extracted from Table 5.3.1.1.2 is provided below:

- TCU (Responsible for wireless connectivity between the device and a cloud service)

- Bluetooth
- ECU (Controls electrical systems in a vehicle)
- OBD-II (Provides diagnostic information on the ECU)
- Steering System
- Break System
- Geolocation data
- Vehicular Sensor Data
- Biometric Data
- Behavioural Data

These constitute a subset of the relevant and potentially vulnerable components in micromobility vehicles. Each micromobility enterprise may utilise a different combination/type of technology, so considerations should be made based on the context of each vehicle design.

6.2.4 Overall Design Considerations

General modelling design considerations were drawn from researchers comments and experiences. First, cross cutting elements are identified as problematic elements in a modelling context [108] [63], this is discussed in more detail in section 6.6.6. The specificity and generality of a modelling language is usually dependent on the types of elements defined within it [107] [108]. For example, the modelling language UML provides many extensions with particular elements which have specific meanings in their own context. This provides superior specificity, with the definition of the model being less burdened with an individuals interpretation of that model. However, it also increases the amount of knowledge one must have about the modelling elements and generally lends itself to less readability due to the increased complexity. ArchiMate provides a different standpoint, offering general elements which can be specialised into various contexts. This provides a smaller language with less elements which is more readily readable, however, a model may be interpreted differently by individuals due to the lack of specificity.

The trade-off between these two attributes, specificity and generality is not concrete, as with a more general language you have the opportunity to provide specific elements in specific contexts to achieve a similar result. The difference is that ArchiMate, and TOG by extension, do not provide these elements.

When designing a security modelling tool, there are further considerations to take into account. A set of these were outlined during the creation of the MVS tool [79] in which the authors propose three relevant considerations.

1. MVS should work on multiple platforms
2. MVS should help enhance the situation awareness of the user during its runtime

3. Node values and types should have intuitive visualisations

Requirement one provides limitations on a proposed solution, and aligns with ArchiMate's ethos of being available on many third party modelling applications. A security solution in this context cannot require new unique behaviour from modelling applications as this would exclude many third-party applications from creating ArchiMate models, reducing its desirability as an EA modelling candidate. Requirement two outlines the utility of the solution, and how its fitness can be measured as how the model translates as awareness to the user regarding the presented system. Finally, requirement three discusses the visualisation aspects of elements, similar to visual accentuation done in [73] and [79].

6.2.5 Conclusions

In conclusion, several aspects of micromobility vehicles and services have been discussed in regards to security modelling. The context of micromobility brings with it unique security and safety requirements. Physical attacks are more likely, and DoS attacks provide an imitate gate between micromobility enterprises and their adoption into the critical transport domain. Elements for the modelling of security in the context of micromobility can be based off previously instantiated elements from research on vehicle modelling and the identification of vulnerable components.

Safety must be taken into consideration when proposing security modelling in a micromobility context due to the increasing risk of physical harm. The joint proposition of safety and security provides quality attributes, that when proposed together, can help consolidate a mitigation solution that promotes security without sacrificing safety and vice versa.

The beginnings of a set of vulnerable micromobility components was presented that identifies a set of components which should be taken into account during security modelling. This set of components should be supplemented with the individual devices technology in order to create a full understanding of its vulnerable components.

A set of general design principles regarding attributes such as specificity, generality, and cross cutting elements were discussed. The modelling of third party solutions needs to be investigated as well as the associated risk and security implications in the context of EA. Any security solution should attempt to retain ArchiMate's ethos of generality and universality, maintaining ArchiMate's ability to be modelled via third party modelling solutions.

6.3 RQ3

What security strategies do micromobility companies currently employ?

This research question helped identify security methodologies, strategies, and mitigation techniques of which a security modelling language would need to support. To answer this, three viewpoints were identified - First, security methods that are used in relevant contexts. Second, technical security and privacy solutions and their descriptions and finally, relevant industry standards that have been utilised in these works in order to provide a basis in industry.

6.3.1 Security Methods

A primary finding in the SLR is the emerging theme of safety and security co-engineering methods which have all been published in recent years [107] [74] [67] [131]. These outline the creation of new co-engineering techniques as well as provide frameworks in order to categorise these techniques.

Three examples of these co-engineering methods can be found in [107], [67], and [131]. Kavallieratos et al. [107] introduces a safety and security combined method, drawing upon the Secure Tropos and STPA to create SafeSec Tropos.

Cui et al., [67] and Riel et al. [131] provide similar methods in contexts parallel to the micromobility context with [67] providing a method of joint safety and security analysis for application in the AV domain and [131] providing an integration method of safety and security for smart products. The increasing rate of research on joint methods and their closely associated application domains identifies these methods as candidates for micromobility enterprises.

These new methods are well categorised within three distinct approach's [107]:

- Security-informed safety approaches
  - Methods that extend safety engineering by enriching themselves with security aspects
- Safety informed security approaches
  - Approaches that extend the scope of security engineering by adapting safety-related techniques
- Combined safety and security approaches
  - Combined approaches for safety and cybersecurity co-engineering

These classifications clearly identify the movement towards the coupling of these two topics, with both methods from the safety and security domains being extended to include the other. These findings show the increasing value in the simultaneous engineering of security and safety quality attributes. A security strategy within the MaaS domain is likely to include aspects of safety, either implicitly through

security policy, or explicitly as quality attributes of the enterprise and vehicle design. Research needs to be done regarding the relationship between security and safety quality attributes to clearly delineate the responsibilities of both, and by doing so, identify what mechanisms are security motivated vs. safety motivated.

One of the primary applications of EA is governance, allowing stakeholders the appropriate information in a timely manner to facilitate decision making. This goal fits well with a risk based method, specifically designed to support governance [27]. In this work the authors surveyed 59 firms regarding their risk processes and information security governance behaviours. They found that 80% of the firms reported the use of cyber risk management methods while 65% of respondents indicated that the board held oversight over information security practices. This indicates an appetite for security governance at the board level, something that EA could be poised to deliver.

Agility in EA practice is a current topic of conversation (see Section 6.1.3) as many EA methods incur significant overhead. Papke, [122] and Solhaug and Seehusen [41] discuss agile security methods and business structure. Papke [122] enables agility through enabling MBSE in collaboration with EA and as such bringing the benefits of an agile work from from MBSE into EA. Solhaug and Seehusen [41] present a method of continuous risk analysis which is enabled by traceability documentation, providing a means to update a model and automatically identify all affected elements. These methods are enabled by MBSE and traceability documentation respectively which may indicate ways of improving EA's documentations agility.

6.3.2 Technical Security and Privacy Solutions

Identifying and understanding types of technical solutions which may be modelled in a micromobility security model enables the design of elements to encompass these applications. Tables 5.17 and 5.23 provide an overview of the extracted privacy and technical solutions respectively while sub-tables 5.24, 5.25, 5.28, 5.26, 5.27 include definitions for technical solutions organised by their respective primary paper.

Technical security solutions offered in the context of the vehicle domain provide the most insight into potential micromobility technical solutions due to their similar application. Two vehicle-centric, privacy technical solutions were described in [110] and [60].

Ming and Yu [110] provide a method of privacy preservation for vehicles which utilise vehicular sensor networks. In using these networks, a vast amount of information is obtained from the environment regarding road conditions and environmental parameters. This data is aggregated and collated in servers which can then be used

to plan routes and trips to locations. Vehicles can query this data in order to plan their own routes, however, in doing so they may disclose their location. The authors use the method *modified oblivious transfer* [199] in order to craft queries which do not disclose the vehicles location.

Xiong and Lagerström [60] utilise a privacy mechanism *Local Differential Privacy* [200] which specifies some architectural constraints and operational rules. The method relies on a form of noise perturbation - adding noise to individuals data in order to maintain a users geolocation data privacy.

Both privacy solutions extracted from the vehicle domain were developed to support the privacy of a users location data, highlighting this as a primary privacy issue. Micromobility enterprise's face similar challenges with increased risk as location data is more abundant in micromobility applications. A private vehicle, such as a car, generally only reports its location when requested to via the user - for example for map directions. Micromobility enterprise vehicles however have many points of continuous GPS recording, through the application the user is using to access the device, the device itself and, potentially a back up GPS device. Micromobility enterprises require geolocation data in order to lower their associated risk of physical damage and theft and also require this data for analytical purposes such as modelling vehicle demand and employee retention (how many employees need to be operating in an area in order to retrieve and charge vehicles). Because of this, geolocation data has become a major enabler of their business models. This creates a conflict between a users privacy, and the utility of geolocation data to the enterprise. A naive solution is the sanitation of data between the vehicles and servers, reducing the data to not include identifiable information. This solution does not account for habitual behaviours and repeated trips which can infer much information regarding a persons behaviours and personal details. A historical example of this is when Uber released a statistical analysis of probable one-night stands [201], inferring this information from a combination of ride and demographic statistics.

Three primary papers provide technical security solutions for application in the vehicle domain [75] [62] [63]. These technical solutions are similar to those found within the CPS, IoT, and General domains and identify general networking security solutions such as host firewalls, static ARP tables and dedicated gateways. More specific technical solutions such as address space layout randomisation which protects against memory attacks (such as No Operation (NOP) sledding) provide specific protections for ECUs (see Section 6.2.3) and other on-board computers.

Revisiting the classification of *Cyber*, *Physical*, and *Social* attack vectors identified in Section 6.2.1, these security technical solutions fit squarely inside of the Cyber domain. As expressed earlier, micromobility vehicles are often subjected to physical attacks, and as such, technical solutions regarding this domain should be explored

more in-depth either through product design or addressing the problem from a software viewpoint in which the firmware contains methods to defend against foreign inputs and attacks.

Generally, the extracted security technical solutions are consistent across domains. Similar networking and device protections are offered across the board highlighting their homogeneous applications.

6.3.3 Industry Standards

Industry standards used in primary papers can be categorised similarly to classifications used in Section 6.3.1 on security methods. These classifications are security based standards, safety based standards, and finally, joint security and safety standards. This further highlights the coupling between security and safety requirements explored within the research in this SLR.

Joint security and safety standards used within works discussing joint security and safety co-engineering methods do not directly apply to the CPS/IoT/AV and connected vehicle application domains. This highlights a discontinuity between current industry standards regarding safety and security, and research domains regarding safety and security. For example, **EN 50126:1999** identifies safety and security requirements for railway applications and was used in work defining safety and security in the context of AV's [76]. This consistent observation indicates that currently, there are few joint security and safety industry standards that apply to these specific research areas.

Comparing these findings to security specific and safety specific industry standards we can see that there are many well aligned standards of which researchers can utilise. **IEC 61508** provides general safety guidelines applicable to IoT and CPS technologies, **ISO 26262** provides functional safety for electronic systems in road vehicles, **ISA/IEC 62443** provides guidelines to mitigate security vulnerabilities in industrial automation, and **ISO 27000** which provides methods of information security management.

It seems that when discussing security and safety on their own there are more standards in general, as well as more applicable standards to the IoT/CPS/AV and connected vehicle domains than when discussing the joint concept of security and safety. As indicated in Section 6.2.2 and Section 6.3.1, joint security and safety processes are becoming increasing popular for technologies such as micromobility vehicles. It would be valuable to see industry standards designed to facilitate these processes.

From all the extracted standards nine were identified to be applicable sources of information for application in a micromobility enterprise context. These include

standards which are applicable to vehicles, as well as the associated applications which users use to interface with the enterprise. These are provided below:

- ISO 21434 [140]
  - Details cybersecurity engineering for application in the vehicle domain.
- ISO 27000 [141]
  - Widely applicable standard defining information security terms, definitions and mythologies.
- ISO/IEC 27034 [142]
  - Standards which provide guidance on information security regarding application systems.
- ISO/IEC 29100 [143]
  - Provides high-level recommendations regarding privacy and the protection of personally identifiable IT.
- SAE J3061 [147]
  - Outlines recommended cybersecurity practice in regards to the vehicle domain.
- ISO 31000 [149]
  - Risk management framework which addresses operational continuity.
- IEC 61508 [155]
  - Addresses functional safety through safety life cycle engineering.
- ISO/PAS 21448 [156]
  - Provides a framework in which intended use and potentially misuse affect a vehicle in combination with a hazardous event.
- ISO 26262 [158]
  - Defines functional safety for electronic systems in road vehicles.

6.3.4 Conclusions

In conclusion, while security strategies are generally not discussed in the direct micromobility context, parallel contexts such as IoT/CPS and connected vehicles provide actionable information on potential security mechanisms and solutions. Security and safety methods (engineering, requirements elicitation, etc.) are becoming a theme in recent research regarding AV's, connected vehicles and, by extension, micromobility vehicles. Governance of risk processes and security mitigation is also a growing requirement as identified in [27] which aligns well with the proposition of security modelling within the EA context.

Geolocation data was identified as the primary concern for vehicular privacy, which provides an interesting discussion on how location privacy can be archived in a service, such as micromobility, which is effectively enabled by such data.

In general, there seems to be a lack, and a need, for joint security and safety standards in industry today. Research which sets out to identify or create new security/safety methods often utilised knowledge from standards who's application domain was not parallel with the goal of the work. It would be beneficial to see joint action between researchers and industry on these topics.

Finally, a list of micromobility enterprise applicable industry standards were provided based upon the set of extracted standards from relevant primary papers. This provides a basis and small review of the available industry literature which could be used in future research to consolidate modelling elements and behaviours.

6.4 RQ4

What support do architectural design languages provide for security in EA?

Architectural design languages provide many security modelling methodologies. These can be classified by their application context to provide useful information in the context of micromobility and EA. First, how to represent attacks, and the visualisation methodologies used to do so, and second, how cybersecurity modelling languages have been designed to provide qualitative and quantitative information for their users.

6.4.1 Representation of Attacks

There are many ways to represent attacks on a system, and depending on the goal of the analyst one may choose an approach which is conducive to formal modelling - extracting objective quantifiable information about an attack from a model, or visual modelling - providing the user a contextual understanding of the attack, its variables, and facilitating the conversation regarding mitigation techniques.

One common attack representation technique is to utilise a tree structure which provides an implicit relationship between a root node and its leaves. For example, the root of an attack tree could be "gaining access to an email account". From this root node, leaf nodes describe what *necessary* conditions must be true in order to achieve the goal. In this way an attack can be represented and enriched to provide measurements (statistical likelihoods of leaf nodes being true) if appropriate data is available.

Fault trees follow a similar method but adopt a different perspective. Fault trees identify how combinations of system failures can lead to undesirable state. This perspective pivot is similar to the current push towards coupling safety and security

engineering methods, attempting to provide a wider scope than just security. Attack-defence trees also seek to provide a wider scope, including mitigation steps and defensive postures into their representation to more accurately identify the current state of the system.

The value of these representations is based with their tree structure which facilitates an overview of the architectural design of a system, and how it may be exploited by an attacker. The created contextual understanding can be used to develop mitigation strategies to address the enterprises needs. From this perspective, EA modelling and attack trees provide the same mechanism, describing the current system and enabling discussion regarding improvements for the next iteration. EA models however (specifically those created with ArchiMate and TOGAF) are not restricted to the same tree design rules as these methods, making them incompatible.

Attack graphs are a set of methods which can be applied more broadly, and are unusually generated, in part, automatically [177]. This can be achieved through network and vulnerability scanners analysing hosts and network configuration to discover multi-step vulnerabilities. This approach lends itself towards quantitative measurement as the resulting graphs can often be extremely complicated, making them difficult to use as a contextual tool.

A common theme in attack graph research is the abstraction and reduction of attack graphs in order to provide a readable output [177] [176]. These methods can both reduce computation time required for statistical models and provide discussion and context regarding the scanned systems, an example of these reductive methods is shown in Figures E.1 and E.2.

6.4.2 Cybersecurity Modelling Languages

Three cybersecurity modelling languages were presented in the SLR. These identify methods researchers have used to model security systems, provide sets of applicable elements and introduce methods of security measurement.

[187] presented SecML, a cybersecurity modelling language designed for application in computer science, electrical engineering and other domains. SecML was based upon UML and SysML combining aspects from both to provide a set of diagrams and models which describe cybersecurity concepts such as misuse-diagrams, sequence diagrams and data interface diagrams.

Cross cutting elements (see Section 6.6.6) have been discussed in the context of security models in architectural descriptions in [45]. In order to provide compartmentalisation and reduce the negative effects of cross-cutting elements on model-driven development the authors provide a method which abstracts security features out of

software modules into their own, self contained module. By doing so, these security features are able to be reused and treated separately from the architecture, providing compartmentalisation and promoting model-driven development.

A primary theme with cybersecurity modelling is the inclusion of probability to enable the identification of risk and likelihood of an attack with mitigation strategies in place (see Section 6.6.3 for more details). [190] and [189] discuss the inclusion of statistical models in their work on the cybersecurity models CySeMoL and P^{2}CySeMoL (which is based on CySeMoL). A common method to achieve this was to utilise a PRM which extend Bayesian networks to include concepts such as objects, relationships, and properties making them ideal for inclusion into modelling language's. P^{2}CySeMoL extends PRM's by utilising the P^{2}AMF method which has superior computation efficiency. A similar extension was provided for ArchiMate in [167] who enabled availability analysis through embedding a PRM into the ArchiMate formalism.

Extending modelling languages to include statistical models for analysis has become reasonably straight forward, however populating these models with the appropriate statistics is difficult for security analysts. There are several different knowledge domains that an analyst may draw from - historical statistical data, expert opinion, risk tolerance – and these have their own associated error probabilities. Methods to identify what statistics are the least error prone and what their associated error probability are would be useful.

6.4.3 Conclusions

In conclusion, there are many distinct security modelling methods used within architectural design languages. Attack trees provide a method of identifying attack paths and promote discussion regarding the mitigation strategy of the enterprise. Attack graphs deliver quantifiable statistics regarding the number of attack paths, their complexity, and the overall risk picture of a system. Attack graphs are often very complex, lending themselves to abstraction in order to be readable.

EA models are often complex in and of themselves, maintaining different abstraction levels based on the architect and their objectives. For example, an architect who is investigating the customer support processes in order to identify improvement opportunities would create a concrete model of these systems. On the other hand an architect who wishes to represent the enterprise in its entirety might produce a more abstract model, which represents customer support with less specific elements. How can a security extension operate alongside these changes in abstraction level?

Statistical modelling is a popular analysis method in cybersecurity modelling, usually utilising PRM's or their modern equivalents. Enabling this analysis pro-

vides a good platform to extract quantifiable information, however identifying what statistics to use and, when statistics are absent, how to represent them is difficult.

6.5 Interview Insights

Interview findings are classified into two perspectives - the industry perspective (Section 6.5.1) and governmental perspectives (Section 6.5.2).

6.5.1 Industry perspective

The industry perspective was gained talking to an enterprise who operates in the vehicle services domain, providing rentable vehicle options to the public. This service utilises an application in order to rent their vehicles and follows the same general business structure utilised by many micromobility enterprises.

Third party security solutions

A primary finding is the application of third party technologies and solutions enabling SME to focus on their business service rather than underlying security implications.

> "One of our big views on security ... is that we actually just outsource a lot of it. Our hardware box and the backing sort of API stack is a software as a service product that we buy from a company in (another country)."

Utilising third party hardware and/or software enables the enterprise to offload a large percentage of the overhead associated with maintaining secure systems. This proves beneficial if the enterprise does not have the resources, or personnel to implement such systems. These findings reinforce those from Section 6.2.1, which identified that modelling solutions need to be developed that can describe these third party systems.

Not all security is outsourced however, as the application which links together consumers and vehicles is developed in house and requires its own security mechanisms.

> "A lot of the technology we're building around [the application] side of the business is proprietary because we think its really actually some of the more valuable stuff and we do do that in house."

An example security mechanism that was introduced in this domain is a facial recognition system designed to disable identity fraud through the provided application.

> "[We implemented a] piece of identity verification which is that we require a selfie now from everybody who signs up."

This system further prevents identity fraud on their application platform. Additional application security primary revolved around data sharing and privacy.

External data sharing, required by government and councils where the enterprise operates, require data to be sanitised - aggregating data to data points such as total number of users, number of trips, and geographical area the trips took place. Internal data sharing follows identified processes in which are designed to reduce the likelihood of data misuse. Methods such as anonymised data exports and the separation of customers from trend statistics are used to reduce privacy risk.

> "...Everything we have is stored in a single cloud provider and we have one copy that lives with our vendor in the Europe region and one copy, which is our copy which lives in the Pacific region. We don't give access internally to anything [similar to that] of big data type ... without that being like logged or like obfuscated in such a way that you couldn't really [enable big data analytics]."

Primary security concerns

Two primary security concerns were identified, identify fraud and physical modification of the provided device.

> "The biggest [security concern] that we have had has been identity theft and the like. ... [As well as] the concept of someone who might have a valid identity with a valid driver's licence and everything, but he was actually just signing up just to take [a vehicle] for a joy ride."

This presents an attack vector (identity fraud) which was not discussed during the SLR, which may be a primary consideration for micromobility enterprises. This highlights some of the disconnect between what research and industry consider to be primary security considerations. In industry this can extend to cases in which individuals misuse the offered service in ways not explored by research.

These cases may not be explored in research due to the fact that these actions may not be negative to the service itself, but are not in line with the enterprises social values.

Physical modification and theft of small micromobility vehicles was also identified as a primary risk, highlighting the fact that it is relatively easy to disable an electric scooter through disconnecting their power sources.

Geolocation Data

Geolocation data was identified as an integral aspect of the services operation, supporting the findings of Section 6.3.2.

> "We do have the ability to monitor where [vehicles] are in real time. Yeah we have reasonable levels of access to that internally. ... We [have a need] to monitor things."

These services require to monitor their vehicles to reduce the risk of many factors such as theft and other criminal activities. Continuous geographical data is uploaded live to service analysts who can monitor, and remotely control aspects of their vehicles in order to protect their investments.

> "Every sort of ten seconds, we have a GPS output of were the [vehicle] is and how fast its going."

General trends in research show a good alignment with industry on the topic of geolocation data (Section 6.6.2) however, perspective often changes between the protection of the asset vs. the consumers.

6.5.2 Governmental perspective

The governmental perspective was gained by interviewing a transport engineer who works in relevant fields in a government department.

Licensing Agreements and Data Sharing

External data sharing is often required in licensing agreements between micromobility enterprises and local governmental bodies.

> "There is a lot of stuff to do with sharing with local councils, and I know from a perspective of our research, we found that [there isn't] necessarily consistent information sharing with those councils about like where micromobility are being used and how often, and by who."

This can become cumbersome if an enterprise operates in many geographical regions, it may need to provide different sets of data for each council as the type and format of information required by each is not consistent.

> "But that's definitely something that we came to encourage to be done more consistently nationally, but obviously that requires data security stuff to be reasonably built at the point you're going to be sharing it."

From the perspective of micromobility enterprises external data sharing is of much more import than other domains due to these governmental licensing requirements. This should be reflected in EA models with modelling tools able to identify data lifecycle and data handling.

Policy, Technology development and Regulation

Regulatory policy on micromobility technology often incurs a lag between the technology being implemented and new policy regarding the technology.

> "Things like micromobility pose quite a challenge in that regard because they are developing faster than policy can keep up."

This can provide a challenge for micromobility enterprises who may need to alter their business model to meet new policy or may need to meet different policy requirements from geographical location to geographical location.

With the quick evolution of micromobility technology comes quickly evolving security considerations. EA and security modelling solutions leveraged in this area need to promote these agile considerations.

> "[micromobilities] evolve at a faster rate compared to the rest of transport - [other transport modes] don't really evolve very fast at all."

Equity and Access

One primary aspect of licensing agreements is the provision of *Equity and Access*. This encompasses the initiative to provide transportation options within disadvantaged demographics. Micromobility enterprises are uniquely equipped to enable this, providing affordable transportation options which can be used for commute and other transportation purposes.

> "Equity and access are [key well-being outcomes] and in some ways, micromobility plays quite well into that"

Disadvantaged communities however, usually exhibit higher levels of vandalism, theft and general crime, increasing the risk probability of damage to micromobility vehicles. This stresses the importance of physical security of these vehicles as outlined in Section 6.2.1.

> "In terms of the security stuff that's a big issue for equity of access, particularly in terms of things getting stolen or broken or any of that. Yeah, so it's a sort of physical issue."

6.5.2.1 Conclusions

Four primary findings impact a potential modelling solution for the EA of micromobility enterprises. Firstly, third party security solutions and technology may be implemented into portions of a micromobility enterprise. As such, methods of including these entities in their EA structure should be implemented.

Second, external and internal data sharing lifecycle's and processes need to be given special attention due to the relationships micromobility enterprises foster with governmental bodies. Modelling solutions should provide clear and distinct elements design to describe these unique processes.

Third, equity of access is a key objective for governing transportation agencies, as such the technology implemented should implement physical security mechanisms which need to be reflected within their models.

Finally, with quickly evolving micromobility technology, policy and legislation will lag behind their applications. This means an agile enterprise structure and modelling approach is more valuable as new restrictions and modifications will become apparent in the future.

These findings are based off a limited number of interviews and as such do not describe the full scale of micromobility enterprise and governmental policy on micromobility enterprises. As such, a future study dedicated to this topic would be valuable and provide a strong direction for future research.

6.6 Overall Observations

This Section serves to highlight dominant themes and pertinent discussion found within the SLR literature. First, safety is discussed due to its strong relationship to security in the vehicle domain. Second, privacy in MaaS industry is discussed. Third, enabling quantitative models through probabilistic modelling is discussed due to the methods prevalence. Forth, architectural automation – and by extension, enterprise agility – is discussed. Fifth, observations regarding Architectural Frameworks and associated layers are provided. Finally, a discussion on cross cutting elements is provided.

6.6.1 Safety as a growing concern

Security research on IoT/CPS technologies, especially when regarding smart transportation, has increasingly become coupled with safety and how these aspects can be treated with respect to each other. This can be seen in the increasing number of recent publications regarding security and safety co-engineering methods [74] [107] [67] [131] of which half investigate the specific context of connected vehicles. Aside from this, other primary papers [76] [171] observe the increasing number of safety and security modelling techniques and, an increasing number of attack description methods which increase their scope to include safety related concepts.

Defining security and safety can be difficult as they are closely joined domains. For example, encryption on phone calls can be thought of as a security mechanism

but it also enables the safety of the users conversations, and by extension the user themselves. In this way safety can be thought of as a goal or quality that security mechanisms aim to provide. The issue with this is that it does not identify situations in which two viewpoints, such as risk to an enterprise and risk to a customer, may be opposing. Take *enterprise x* who offers an electric scooter for rent in *zone a*. If a user rides this scooter out of *zone a* what behaviour should the security mechanism, designed to stop theft and maintain zoning exhibit? In regards to the physical safety of the individual certain behaviours can be ruled out. In this way safety is used as a framework in order to answer potentially ambiguous questions regarding the security of the enterprise and the safety of the customer.

By including safety in the discussion of security provides a way to identify security mechanisms that promote the safety of the user while protecting the enterprises investment. As it stands, micromobility enterprise would benefit from a joint security/safety approach due to its deployment into public spaces and its inherent risk of physical harm. Because of this, a security extension to ArchiMate for use in EA should include aspects of both security and safety to enable these decisions to be reflected in an EA documentation.

6.6.2 A new understanding of Privacy

With the advent of data hungry applications and technology privacy has quickly become a popular topic in both the research and public domains. IoT and CPS technology is often categorised as an enabler of these data hungry applications ranging from vehicular sensor networks reporting on road and environmental conditions to smart speakers, bulbs, coffee machines and beyond. In this context, privacy is an issue of covert surveillance [46] [57] [58] in which information is unknowing gathered and utilised.

Privacy specific modelling elements have been proposed in research [60] [57] which identify types of sensitive data (geolocation data, vehicular sensor data, behavioural data) and identifies methods which promote consumer privacy. Other works, such as [57] and [59], provide guidelines to help organisations align themselves with governmental privacy regulation and identify compliance benefits which can be born from this [58].

Micromobility applications contain many points of interest from a privacy perspective. Primary aspects are geolocation data, data life cycles, data storage and aggregation and, finally user and demographic data. As discussed in Section 6.3.2, geolocation data is a primary enabler of the micromobility enterprise business model, providing information for distribution, vehicle recovery, non-repudiation, and other

applications. Because of this, methods such as optimising data lifecycles to promote privacy, data aggregation, data sanitation, and customer informing techniques (Terms of Service, etc,.) become an important aspect of their operation. Because of this, a security modelling extension should support the modelling of these processes, outlining the transformation data goes through to provide a clear understanding of any vulnerabilities there are in the enterprises approach to privacy.

6.6.3 Probability and modelling

A consistent theme throughout security formal modelling research is the inclusion of probability models in order to derive probability metrics (see Sections 6.1.2 and 6.4.2). Five primary papers enable probability analysis, [175], [178], [189], [190] and, [167].

Three of these papers implement PRM's or extensions of PRM's (P^2AMF). [189] integrated PRM into the cybersecurity language CySeMoL in order to analyse enterprise system architectures in regards to attack success rates. This was then extended in [190], who interrogated P^2AMF as a replacement for the previous PRM architecture to enable better computational performance. Finally, [167] utilised a PRM to enable availability analysis on fault tree structures (Section 6.4.1 on fault tree structures).

Other than PRM's, researchers have used base Bayesian networks and other methods to encode probability statistics into their models. Kordy et al. [175] implements a form of probability evaluation through encoding Bayesian networks into AD-trees in order to evaluate probabilistic factors. Khouzani et al. [178] encodes probability by associating a probability metric with each edge in an attack graph with the factor representing the success rate of an action occurring.

It is clear that using probability modelling is a popular choice within security modelling, enabling analysis to manage security risk by identifying likely attacks, providing targets for mitigation within an organisation. Its weaknesses stem from the knowledge bases of which these probabilities are taken when informing the model. Each probability factor has its own probability regarding the amount of trust the analyst has in it accurately representing attack statistics. For example, historical statistics regarding DDoS attacks from 2010 would not be relevant in today's ecosystem. Because of this it is important to also measure the error of the probabilistic model, acknowledging the potential risk of using an out of date model.

Including probability analysis into security modelling in the context of EA provides an interesting question. Is it feasible to provide an extension to the ArchiMate language which includes a self-contained probabilistic model for analysis while being included into the overall architectural description? Researchers have already

provided an ArchiMate extension including a probabilistic model ([167], see Section 6.1.2) targeted at availability analysis rather than security analysis. This solution however requires a new model to be built specifically for the investigation of availability, it does however indicate that integrating probability models into ArchiMate is possible and newer research on this topic could yield good results.

6.6.4 Automation of architectural descriptions

Automation provides a method of reducing the work required when describing an architecture. In this way, automation encourages EA agility (Section 6.1.3), increasing the speed that an architecture can be investigated. Automation methods were discussed in Section 6.3.1, where [41] automatically identifies elements in an architecture which must be reevaluated during risk analysis, and in Section 6.4.1 on attack graphs, which utilise automated modelling to create formal models of an enterprises security posture.

Two primary papers specifically provide automation methods for use in EA. [184] designed the tool *Automatic Designer* which is able to produce architectural solutions designed to provide security analysis. [182] designed a tool which automatically gathers the data required to create enterprise models. To do this they utilise vulnerability and network scanners. These papers, and all other primary papers which research automation, are relatively older in comparison to the majority of included works. This could be due to a limitation regarding the applicability of each method to specific enterprise contexts. EA, as a practice, is not standardised. The tools and methods used change from architect to architect which makes it difficult to provide an automation solution, which are usually quite ridged in terms of required input and expected output, that is applicable to these different contexts.

Security modelling, however, does lend itself to automated modelling. Specifically, cyber security modelling (see Section 6.2.1) is able to be automated through network and vulnerability scanners. This was utilised in [182] and underlines why automated processes are popular during the creation of attack graphs (Section 6.4.1). Physical and social security are more difficult domains to map as they are not classically defined as security vulnerabilities, rather they are design aspects related to security.

Automated safety modelling is more difficult than security modelling due to its wider and more abstract scope. Falessi et al. [185] provides an interesting work which extracts safety relevant slices from an architecture to aid in safety auditing. This is achieved by using safety traceability mechanisms where each element or process has a related safety requirement. These safety requirements can then be used to extract each module that requires safety consideration.

Automation proves to be a difficult tool to implement in an EA context as architects are often trying to elicit specific information from a model, and frame the model to provide this. Cyber security modelling specifically however lends itself to automated modelling, providing an opportunity for a joint approach - joint EA design and automated security modelling. This method would allow for the flexibility that EA requires while providing a standardised security modelling approach which can be implemented in security conscious EA applications.

6.6.5 Architectural Frameworks and Structure

During the SLR, IoT and CPS technology architectures were extracted to identify different perspectives researchers and industry were using in their approaches. In total 16 IoT architectures and nine CPS architectures were identified. These architectures are divided into categories based on how many layers they define.

Baloyi and Kotzé [46] identified that IoT architectures could often be categorised as instantiations of CPS architectures which would indicate that their architectures should be reasonably similar. Three layer CPS/IoT architectures express this, with IoT architectures containing elements of CPS architectures. For example the CPS layer *Devices* is analogous to the IoT layer *Sensing*. This pattern of analogous terminology is common in frameworks with few layers.

As the granularity of the architecture increase by adding layers these similarities become more difficult to identify. This is due to their application contexts having more influence over each layer. For instance, data management in CPS applications is contextually more important than IoT applications, and as such is included as a layer in their technology architecture.

Similar effects can be seen within EA structure, however, as businesses are more diverse than CPS/IoT technologies this effect is augmented with many different architecture structures being plausible. Because of this, standardisation has been a valuable asset in EA, with, for example, TOGAF providing a set of layers and their descriptions for use in a business context. This can also serve as a limitation as discussed in Section 6.6.6 in which viewpoints are unable to be fully realised with the provided layer structure.

Observations have been made regarding the lack of security-centeric layers in IIoT architectures [89], identifying security as an incompatible layer with the current architectural perspective of IoT and CPS technologies. These limitations arise from the chosen architectural drivers (what goal is the architecture achieving, Section 6.6.6) and the architectural multi-layer framing that is prevalent today.

6.6.6 Architectural Drivers and Cross Cutting Elements

A primary theme and issue facing security modelling in EA and *layered* modelling frameworks is that of cross-cutting elements. Cross-cutting elements are elements or layers which run perpendicular to the provided layer framework. Good examples of this can be found within the primary papers highlighted in RQ1. In Figure D.1 the cross cutting element *security layer* is shown clearly as a perpendicular element to the Business, Application, Technology and Physical layers. Figure D.2 provides a similar example however with the assurance layer rather than security.

Multi-view modelling primarily addresses complexity, compartmentalising the structure of a domain into vertically interacting components. This provides a method of visualisation, decomposition and shows a perspective of the domain which the architect deems the most useful. Multi-view modelling however, can obscure important components of a domain due to the chosen perspective. One such component is security, which if implemented in a model along-side traditional EA frameworks, is either dispersed throughout the model [45] or presented as a cross cutting function of the model [169] [165].

This effect is best described [108] where architectural views are associated to their underlying architectural drivers (their motivations). When an architect designs an architectural framework they identify architectural views (layers) which promote the underlying motivation for providing the framework. This results in a certain architecture structure which enables the task which the architect is trying to perform. EA's underlying motivation is to describe an enterprises primary components and structure, reflecting stakeholders needs and enabling the business to address their business issues and needs. ArchiMate, TOG's visual modelling language for EA, exhibits this motivation in the form of five layers (ArchiMate full framework, 6.1). Each layer provides infrastructure and value for the next with good consistency - that is each layer has a very little degree of potential overlap.

ArchiMate's full-framework also specifies a set of vertical elements, *aspects*. The first three aspects, *Passive structure*, *Behaviour* and *Active Structure* are not classified as cross-cutting elements as they classify and organise elements within the five identified layers. The *Motivation* aspect most closely resembles a cross cutting element however constitutes an element with good consistency and does include elements or concepts from within other layers. Compare this to security, an aspect of an enterprise which can be instantiated within at least three of the current layers (Implementation & Migration, Technology and Application layers). This leads to the current problem in regards to implementing security in EA modelling, specifically in the ArchiMate language.

If an architect wish's to provide an emphasis on the security of their domain (such

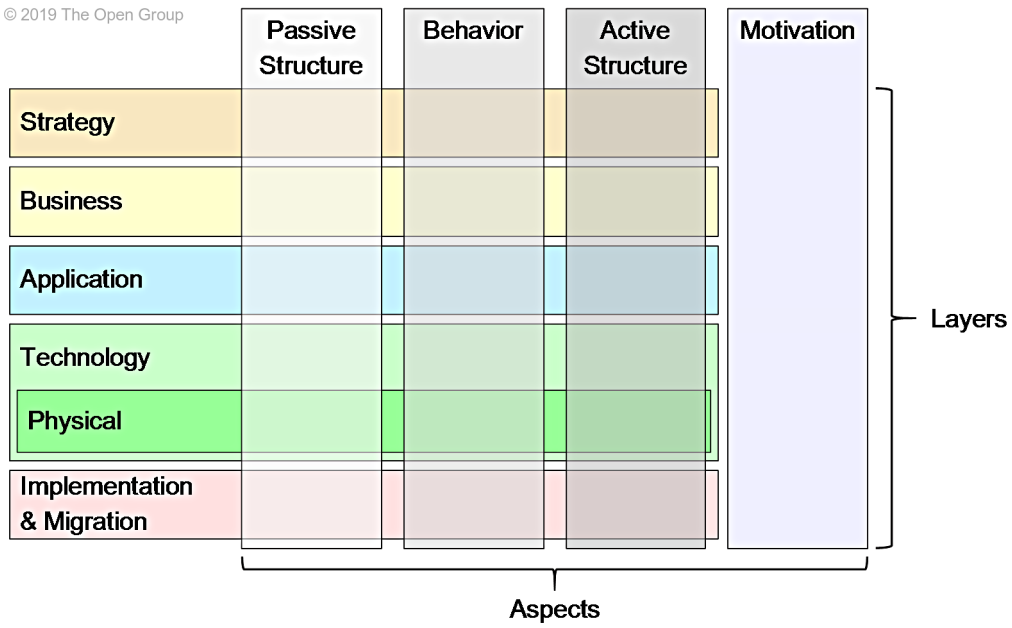


Figure 6.1: ArchiMate full framework (extracted from [2])

as micromobility enterprise) there is very little ArchiMate can offer them in terms of value, leading them to utilise separate modelling methods to describe their security posture. The introduction of the RSO provides a set of elements and relationships designed to provide utility in security modelling which works well in isolated models, however, as an integrated part of ArchiMate's overview model, constitutes a complex cross-cutting element relating multiple layers together.

This lack of consistency can affect the agility of ArchiMate's approach, as well as disabling the ability to present security as a primary consideration to stakeholders in the context of the overall enterprise. This highlights a discontinuity in the multi-view modelling approach for application in EA - EA provides governance on systems and processes deemed contextually relevant to an enterprise's success however does not support viewpoints such as security being modelled together with business processes. With security becoming increasingly important in an enterprise operation an important question needs to be answered: How can security be represented alongside traditional EA framework structure in order to provide insight and oversight for stakeholders?

This question can be extended to other relevant concepts such as safety, privacy, social responsibility and environmental viewpoints. These viewpoints, while important to different enterprises depending on their context, all constitute aspects that an architect may want to accent in their architecture which are cross-cutting and lack consistency. Solutions discussed during the SLR either disregard inconsistencies or provide security modelling without the wider context of EA (even works utilising ArchiMate, a modelling tool for EA) during their presentation, as such new solutions need to be investigating taking into account the issues mentioned above.

6.7 Future Research

The SMS presented in this research has already been published in SEAA [1] and the provided SLR is planned to be published mid 2021. Future research opportunities identified through this research are provided below.

Addressing the cross cutting concerns discussed in Section 6.6.6 one possible research direction is the encoding of cross cutting viewpoints (security, safety) as parameters into elements constituting the model, thus providing a way to view the model from different dimensions. While these dimensions are not visually encoded into the model, they provide the traceability needed to transform the model during stakeholder interactions to present the topic under question. Consistency in this context is extended into two attributes, visual consistency and virtual consistency where virtual consistency identifies how many dimensions the element belongs to. The encoding process may be implemented in many ways - Boolean expressions or discrete values which identify the strength of an elements relationship to the represented dimension, allowing for even more flexibility during analysis.

The modelling concepts instantiated within ArchiMate are motivated from aspects of TOGAF, as it was developed to provide utility to the overall architecture framework. As such, research investigating the architecture frameworks themselves (E.g TOGAF, SABSA) may provide information on how systemic security support may be integrated within the underlying framework. This possible method would be useful in conjunction with the dimensional modelling solution described above. Together, these research topics would provide a stable security modelling solution with structural support from the architecture framework and tooling support from ArchiMate.

The relationship between safety and security is not well defined within both academic research and industry standards. As such, it can be difficult integrating these concepts into models which may benefit from including both perspectives. Security modelling of micromobility, and MaaS industries would benefit from research investigating the interplay between security considerations/mechanisms and safety considerations/mechanisms, providing a framework to identify how safety concerns motivate security and safety quality attributes.

Further interviews within industry and government need to be performed in order to align research to current trends. This would be particularly useful for the micromobility and MaaS contexts due to the lack of industry standards associated with them. The context of EA also needs to be more thoroughly explored from the perspective of industry. Many tools and methods have been proposed in ArchiMate for use in EA, but the uptake and general utility of these extensions are unknown

in industry. Larger interview studies in these areas would provide the required understanding and contextualisation for future research.

There is an appetite in academia to enable quantitative modelling through integrating probability models within ArchiMate. Further interviews would indicate the value in this approach, and if there is also an appetite in industry, potential solutions which integrate EA modelling and probabilistic models should be done. Integrating probabilistic models into EA models is complicated by the fact that core elements in an EA model may not have statistical information regarding them (either due to the nature of the element, or due to a lack of historical data, for example). Research discussing how incomplete probabilistic models, in the context of EA, would be a valuable addition to this area.

Chapter 7

Conclusion

This research was designed to investigate the intersection between micromobility enterprise, EA, and security modelling. Micromobility, as a research topic is becoming increasing desirable as a growing number of MaaS enterprises begin operating around the world. These transport alternatives are driven by recent advances in IoT, CPS and connected vehicle technologies, providing the risk mitigation needed to enable these business models. With these quick advances however, comes a lag in domain specific research, industry standardisation, and governmental policy providing a valuable opportunity to investigate this unique context.

To answer the research question – "How can EA provide security modelling utility in a micromobility context?" – a three component methodology was designed, consisting of an SMS study, an SLR study, and industry interviews. This methodology was selected due to a confluence of factors: micromobility specific research was sparse, reviews provide the flexibility required when investigating new contexts, and these methods provide a strong foundation for future research. The SLR study constitutes the primary contribution of this research.

In total, the SMS returned 13 primary papers and provided feedback on candidate SLR method configurations. Using these findings the SLR was modified and run, identifying a total of 71 primary papers for synthesis. The SLR synthesis was enriched with three interviews from government and industry sectors. These results, and discussions of future work, are provided in Chapter 6. Four research questions were designed for the SLR identifying the current state-of-the-art methods for security modelling in ArchiMate (RQ1), micromobility security aspects (RQ2), micromobility security strategies (RQ3) and finally, architectural design and security modelling support (RQ4). These research questions cover primary areas of knowledge in terms of micromobility security and security modelling.

RQ1 identified a set of research which provides security modelling methods in ArchiMate. Two general methods were utilised when providing security modelling

in ArchiMate. First, the reuse of existing ArchiMate elements and relations to enable security modelling and second, the creation and addition of new elements and relations to support security modelling. Two issues arose in regards to it, first, the application of the provided security solutions were very contextual, and therefore were unable to be applied outside of those contexts. Second, the security solutions were often not designed with regard to EA, making it difficult to migrate their solutions to an EA context.

RQ2 provided two primary insights - micromobility applicable attack vectors defined in research and, previously utilised and instantiated security modelling elements developed in similar application domains. These insights simultaneously identify real world attack contexts in regard to micromobility and methods of modelling these. Overall, while cyber attacks are the majority of the identified attack vectors micromobility uniquely diverges from general IoT/CPS applications with an emphasis on physical security. Social security consisted of the least research, however, during interviews new security considerations such as identity fraud were identified as a primary security consideration. The majority of extracted modelling elements were defined in parallel application contexts such as the connected vehicle domain, modelling connected vehicles/AV. As micromobility vehicles are a subset of connected vehicle/AV vehicles, some of these elements are applicable within the micromobility domain.

RQ3 identified security strategies, solutions and standards used to mitigate attack vectors described in RQ2, and provides information on what processes may need to be modelled when describing the security posture of a micromobility enterprise. A set of security methods, technical solutions and micromobility applicable industry standards were extracted from research. A primary finding is the discussion and development of closely coupled security and safety engineering methods for application in vehicular contexts. This aligns well with micromobility enterprise due to the physical service they offer, elevating the importance of safety in their application. This theme is extended to industry standards, with many safety and security standards being used to educate engineering methods. These standards however were developed in unrelated fields, which outlines the current lack of standardisation for safety and security engineering in the smart vehicle domain. The identified technical solutions provide a set of possible solutions that an architect may wish to represent in their model, and act as a method for measuring the coverage of a proposed security modelling solution.

RQ4 discussed security modelling methods and contributed three primary insights into attack/defence modelling, architectural automation and cybersecurity modelling languages. Two primary forms of attack representations were categorised - attack trees and graphs. These have been researched extensively and provided methods eliciting both qualitative (attack trees) and quantitative information (at-

tack graphs). Many of the discussed cybersecurity modelling languages implement a form of attack graph, enabling the quantitative analysis of attack scenarios - usually through probability metrics. Architectural automation and security modelling showed unexpected correlation as many architectural processes used security vulnerability scanners in order to map and model enterprise systems. These findings explore security representation and modelling, providing information on current practices, and outlining methods which achieve the desired security modelling outcomes.

7.1 Future Work

Future research should focus on addressing the core issue of cross-cutting security elements, and more generally, how to include cross-cutting domains (security, safety, environmental) into EA structures. This research needs to understand both the modelling tools – ArchiMate – and the underlying architecture frameworks in order to provide systemic reflections of these cross-cutting domains.

Periphery future work includes defining the relationship between the concepts of security and safety – a definition that is becoming increasingly important as complex technology becomes increasingly responsible for physical and cyber safety. A definition, providing clarification on which motivation an element in a model (say an IDS) should fall under (safety or security) would be valuable for industry professionals, researchers, and policy makers.

Further research reflecting the current state of industry needs to be performed. Currently there is little information on what EA methods are being utilised, and how practitioners are using the provided tooling (ArchiMate). Interviews within industry need to be performed to align academic research with industry.

Finally, a future research opportunity regards integrating quantitative modelling into qualitative modelling methods. How can quantitative probabilistic models be included within a tool such as ArchiMate without increasing the complexity of the model? Further, how can you provide an incomplete statistical model, and still provide useful – and not misleading – metrics?

7.2 Final thoughts

In conclusion, four primary contributions are made by this research. First, a mapping of current micromobility research is provided through the SMS study – outlining the current lack of micromobility specific research, a direct consequence of the technologies quick integration and development. This lack of micromobility research outlines the second contribution: the initial identification and definition of a micromobility security landscape through identifying relevant security considerations from parallel

and intersecting domains such as IoT/CPS/AV/connected vehicle. The third contribution of this research is the insight gained through interviews with industry and government, enriching the understanding of MaaS and micromobility enterprise. Finally, the primary contribution of this research is the SLR, identifying and discussing the current state of security modelling in EA and micromobility security concerns. The SLR also lends itself to identifying gaps in research and future research opportunities.

Acronyms

5W1H Who, What, When, Where, Why and How.

AD Axiomatic Design.
AD Trees Attack/Defence Trees.
AI Artificial Intelligence.
API Application Programming Interface.
APP Acronym undefined in referencing article.
APPI African Protection of Personal Information act.
ARP Address Resolution Protocol.
ARP* Aerospace Recommended Practice.
ATM Air Traffic Management.
ATSyRA Attack Tree Synthesis for Risk Analysis.
AV Autonomous Vehicle.
AVES Automated Vehicles safety and security analysis framework.

BLE Bluetooth Low Energy.
BPM Business Process Management.
BPNN Back Propagation Neural Network.
BPRIM Business Process Risk Integrated Method.

CA-ISFAM Cluster Adapted Information Security Focus Area Model.
CAD Computer Aided Design.
CAN Controller Area Network.
CAPEC Common Attack Patterns Enumeration and Classification assurance strategic initiative.
CHASSIS Combined Harm Assessment of Safety and Security for Information Systems.
CIA Confidentiality, Integrity, Availability.
CPS Cyber Physical Systems.
CRAF Cyber Risk Assessment Framework.
CS Cuckoo Search.
CWE Common Weakness Enumeration.
CySeMoL Cyber Security Modelling Language.

DA Dominator Analysis.
DBaaS Data Base as a Service.
DDoS Distributed Denial of Service.
DISCOA Distributed Coordinated Adaptations.
DoDAF Department of Defence (US) Architecture Framework.
DoS Denial of Service.
DS Database Security.
DSL Domain Specific Languages.
DSM Design Structure Matrix.
DTLS Datagram Transport Layer Security.

EA Enterprise Architecture.
ECSS European Cooperation for Space Standardisation.
ECU Electronic Control Unit.
eEPC extended Event-driven Process Chain.
EN European Standards.
ERM Enterprise Risk Management.
ES-C2M2 Electricity Subsector Cybersecurity Capability Maturity Model.
EUC Equipment Under Control.
EV electric vehicle.

FACT Failure Attack Countermeasures.
FAMM Focus Area Maturity Model.
FEAF Federal Enterprise Architecture Framework.
FEC Fog Edge Computing.
FECIoT Fog Edge Computing Internet of Things.
FR Functional Requirements.
FRAM Functional Resonance Analysis Method.

GPS Global Positioning System.
GSM Global System for Mobile communications.

HSM Hidden Structure Method.

IACS Industrial Automation and Control Systems.
ICAMP IoT and CPS Architecture based Model for data Privacy.
IDPS Intrusion Detection and Prevention System.
IDS Intrusion Detection System.
IEC International Electrotechnical Commission.
IIoT Industrial Internet of Things.
IMSA Intra Model Security Assurance.
IoT Internet of Things.
ISA International Standards on Auditing.

ISFAM Information Security Focus Area Model.
ISO International Organisation for Standardisation.
IT Information Technology.

LAN Local Area Network.
LDP Local Differential Privacy.
LINDDUN Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of the information, Unawareness, Non-compliance.
LTE-A Long Term Evolution Advanced.

MaaS Mobility as a Service.
MAL Meta Attack Language.
MARIA Model of ATM Reality in Action.
MBSE Model Based System Engineering.
MCDA Multi-Criteria Decision Analysis.
MITM Man In The Middle.
MVS Metrics Visualisation System.

NFC Near Field Communication.
NICE National Initiative for Cybersecurity Education Capability Maturity Model.
NIST National Institute of Standards and Technology.
NOP No Operation.

O-ISM3 Open Information Security Management Maturity Model.

P2AMF Predictive Probabilistic Architecture Modelling Framework.
P2CySeMoL Predictive Probabilistic Cyber Security Modelling Language.
PAS Publicly Available Standardisation.
PPSA Privacy Preserving Subset Aggregation.
PRM Probabilistic Relational Model.

R-BPM Risk aware Business Process Management.
RA Risk Assessment.
RAMS Reliability, Availability, Maintainability, Safety.
RDF Resource Description Framework.
RFID Radio Frequency Identification.
ROI Return On Investment.
RSO Risk and Security Overlay.

SABSA Sherwood Applied Business Security Architecture.
SADT Structured Analysis and Design Technique.
SAE Society of Automotive Engineers.
SARA Security Automative Risk Analysis.
SARSA State Action Reward State Action.

SCADA Supervisory Control And Data Acquisition.
SDL Security Development Lifecycle.
SEAA Software Engineering and Advanced Applications.
SecML Security Modelling Language.
SFA Signal Flow Analysis.
SIG Soft goal Interdependence Graph.
SLR Systematic Literature Review.
SME Small, Medium Enterprises.
SMS Systematic Mapping Study.
SOTIF Safety Of The Intended Functionality.
SQL Structured Query Language.
SQuaRE Systems and Software Quality Requirements and Evaluation.
SSL Secure Socket Layer.
STPA Systems Theoretic Process Approach.
STRIDE Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege.
SV Security Vulnerability.
SysML Systems Modelling Language.

TaaS Transport as a Service.
TCP Transmission Control Protocol.
TCU Telematics Control Unit.
TLS Transport Layer Security.
TOG The Open Group.
TOGAF The Open Group Architecture Framework.
TREsPASS Technology-supported Risk Estimation by Predictive Assessment of Socio-technical Security.

UAF Unified Architecture Framework.
UDP User Datagram Protocol.
UFoI-E Uncontrolled Flows of Information and Energy method.
UWB Ultra Wide Band.

Bibliography

[1] A. Ellerm and M. E. E. Morales-Trujillo, "Modelling Security Aspects with ArchiMate: A Systematic Mapping Study," *SEAA*, pp. 577–584, 2020. DOI: 10.1109/seaa51224.2020.00094.

[2] The Open Group, *The Open Group Standard: ArchiMate 3.1 Specification*. 2019, pp. 1–181, ISBN: 9789087536794. [Online]. Available: https://pubs.opengroup.org/architecture/togaf9-doc/arch/.

[3] Wikipedia, *Micromobility - Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Micromobility (visited on 03/30/2021).

[4] SAE, *J3194: Taxonomy and Classification of Powered Micromobility Vehicles - SAE International*, 2019. [Online]. Available: https://www.sae.org/standards/content/j3194_201911/ (visited on 03/10/2021).

[5] Micromobility Industries, *The Micromobility Landscape — Micromobility Industries*. [Online]. Available: https://micromobility.io/blog/the-micromobility-landscape (visited on 03/10/2021).

[6] Micromobility Industries, *Micromobility, An Introduction — Micromobility Industries*. [Online]. Available: https://micromobility.io/blog/2019/1/21/micromobility-an-introduction (visited on 03/10/2021).

[7] Micromobility Industries, *The Three Eras of Micromobility — Micromobility Industries*. [Online]. Available: https://micromobility.io/blog/2019/4/29/the-three-eras-of-micromobility (visited on 03/10/2021).

[8] TomTom, *Traffic congestion ranking — TomTom Traffic Index*. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking/ (visited on 03/10/2021).

[9] U.S. Department of Transportation, *Mobility on Demand Operational Concept Report*. September. 2017, ISBN: 1700000152.

[10] Micromobility Industries, *If You Care About Climate Change, You Should Care About Micromobility — Micromobility Industries*. [Online]. Available: https://micromobility.io/blog/2019/10/22/if-you-care-about-climate-change-you-should-care-about-micromobility (visited on 03/10/2021).

[11] Quartz, *How long does a scooter last? Less than a month, Louisville data suggests — Quartz*. [Online]. Available: `https://qz.com/1561654/how-long-does-a-scooter-last-less-than-a-month-louisville-data-suggests/` (visited on 03/10/2021).

[12] E. Wiles, *Which E-Scooter Model Does Bird Use*. [Online]. Available: `https://www.scootered.co.uk/blog/what-scooter-model-does-bird-use.html` (visited on 03/28/2021).

[13] Neuron, *N3 e-scooter — Neuron Mobility*. [Online]. Available: `https://www.rideneuron.com/n3-e-scooter/` (visited on 03/10/2021).

[14] Ramboll, "ACHIEVING SUSTAINABLE MICRO-MOBILITY," no. April, 2020.

[15] Zimperium, *Xiaomi Scooter Hack Enables Dangerous Accelerations & Stops for Unsuspecting Riders*, 2019. [Online]. Available: `https://blog.zimperium.com/dont-give-me-a-brake-xiaomi-scooter-hack-enables-dangerous-accelerations-and-stops-for-unsuspecting-riders/` (visited on 03/10/2021).

[16] Deloitte, *Addressing cybersecurity challenges in the future of mobility — Deloitte Insights*. [Online]. Available: `https://www2.deloitte.com/us/en/insights/focus/future-of-mobility/cybersecurity-challenges-connected-car-security.html` (visited on 03/10/2021).

[17] S. H. Spewak, *Enterprise architecture planning*. New York, NY, USA: Wiley, 2001, ISBN: 978-0471599852.

[18] U. Congress, "Clinger-Cohen Act," *Public Law*, vol. 1996, no. 5, pp. 1–55, 1996. [Online]. Available: `https://www.treasury.gov/privacy/Documents/Clinger-Cohen_Act_of_1996.pdf`.

[19] S. Kotusev, "The History of Enterprise Architecture: An Evidence-Based Review," *Journal of Enterprise Architecture*, vol. 12, no. 1, pp. 29–37, 2016.

[20] A. Bobkowska, "Modeling pragmatics for visual modeling language evaluation," *ACM International Conference Proceeding Series*, vol. 127, pp. 75–78, 2005. DOI: `10.1145/1122935.1122950`.

[21] P. Hall, C. Heath, L. Coles-Kemp, and A. Tanner, "Examining the Contribution of Critical Visualisation to Information Security," in *Proceedings of the 2015 New Security Paradigms Workshop*, ser. NSPW '15, New York, NY, USA: Association for Computing Machinery, 2015, pp. 59–72, ISBN: 9781450337540. DOI: `10.1145/2841113.2841118`. [Online]. Available: `https://doi.org/10.1145/2841113.2841118`.

[22] A. Watson, "Visual Modelling: past, present and future Vice-President and Technical Director Object Management Group," pp. 1–6, 2017. [Online]. Available: `http://www.uml.org/Visual_Modeling.pdf`.

[23] University Tecnólogica de la Mixteca, *Concept: Visual Modeling*. [Online]. Available: http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/visual_modeling_2C089766.html (visited on 03/11/2021).

[24] Y. M. Yashchyshyn, "Disciplined Heterogeneous Modeling," *Microwave Radar and Wireless Communications (MIKON), 2010 18th International Conference on*, pp. 273–287, 2010, ISSN: 1744-618X. DOI: 10.1111/j.1744-618X.2010.01158.x.

[25] I. Band, W. Engelsman, C. Feltus, S. G. Paredes, J. Hietala, H. Jonkers, and S. Massart, "Modeling Enterprise Risk Management and Security with the ArchiMate Language," *Open group*, p. 40, 2014.

[26] T. Prince Sales, J. P. Andrade Almeida, S. Santini, F. Baiao, and G. Guizzardi, "Ontological analysis and redesign of risk modeling in archimate," *Proceedings - 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference, EDOC 2018*, no. October, pp. 154–163, 2018. DOI: 10.1109/EDOC.2018.00028.

[27] C. Griffy-Brown, D. Lazarikos, and M. Chun, "Agile Business Growth and Cyber Risk:" *2018 IEEE Technology and Engineering Management Conference, TEMSCON 2018*, no. Figure 1, pp. 1–6, 2018. DOI: 10.1109/TEMSCON.2018.8488397.

[28] G. Schryen, G. Wagner, and A. Benlian, "Theory of knowledge for literature reviews: An epistemological model, taxonomy and empirical analysis of IS literature," *2015 International Conference on Information Systems: Exploring the Information Frontier, ICIS 2015*, pp. 1–22, 2015.

[29] B. Kitchenham, "Procedures for Performing Systematic Literature Reviews," *Annals of Saudi Medicine*, 2004, ISSN: 09754466. DOI: 10.5144/0256-4947.2017.79. [Online]. Available: http://www.annsaudimed.net/doi/10.5144/0256-4947.2017.79.

[30] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015, ISSN: 09505849. DOI: 10.1016/j.infsof.2015.03.007. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2015.03.007.

[31] E. Mourão, J. F. Pimentel, L. Murta, M. Kalinowski, E. Mendes, and C. Wohlin, "On the performance of hybrid search strategies for systematic literature reviews in software engineering," *Information and Software Technology*, vol. 123, no. July 2019, 2020, ISSN: 09505849. DOI: 10.1016/j.infsof.2020.106294.

[32] K. Louise Barriball and A. While, "Collecting data using a semi-structured interview: a discussion paper," *Journal of Advanced Nursing*, vol. 19, no. 2, pp. 328–335, 1994, ISSN: 13652648. DOI: 10.1111/j.1365-2648.1994.tb01088.x.

[33] ISO/IEC, "Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models," International Organization for Standardization, Standard, 2011.

[34] B. Kitchenham and S. Charters, "Procedures for performing systematic literature reviews in software engineering," *Keele University & Durham University, UK*, 2007.

[35] M. Korman, T. Sommestad, J. Hallberg, J. Bengtsson, and M. Ekstedt, "Overview of Enterprise Information Needs in Information Security Risk Assessment," *Proceedings . IEEE 18th international Enterprise Distributed object computing conference*, vol. 2014-Decem, no. December, pp. 42–51, 2014, ISSN: 15417719. DOI: 10.1109/EDOC.2014.16.

[36] M. Korman, R. Lagerström, and M. Ekstedt, "Modeling authorization in enterprise-wide contexts," *CEUR Workshop Proceedings*, vol. 1497, pp. 81–90, 2015, ISSN: 16130073.

[37] N. Mayer, J. Aubert, E. Grandry, C. Feltus, E. Goettelmann, and R. Wieringa, "An integrated conceptual model for information system security risk management supported by enterprise architecture management," *Software and Systems Modeling*, vol. 18, no. 3, pp. 2285–2312, 2019, ISSN: 16191374. DOI: 10.1007/s10270-018-0661-x. [Online]. Available: https://doi.org/10.1007/s10270-018-0661-x.

[38] M. J. Anwar and A. Q. Gill, "A review of the seven modelling approaches for digital ecosystem architecture," *Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019*, vol. 1, pp. 94–103, 2019. DOI: 10.1109/CBI.2019.00018.

[39] T. Pavleska, H. Aranha, M. Masi, E. Grandry, and G. P. Sellitto, *Cybersecurity Evaluation of Enterprise Architectures: The e-SENS Case*. Springer International Publishing, 2019, vol. 1, pp. 226–241, ISBN: 9783030351502. DOI: 10.1007/978-3-030-35151-9_15.

[40] M. Shariati, F. Bahmani, and F. Shams, "Enterprise information security, a review of architectures and frameworks from interoperability perspective," *Procedia Computer Science*, vol. 3, pp. 537–543, 2011, ISSN: 18770509. DOI: 10.1016/j.procs.2010.12.089.

[41] B. Solhaug and F. Seehusen, "Model-driven risk analysis of evolving critical infrastructures," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 2, pp. 187–204, 2014, ISSN: 18685145. DOI: 10.1007/s12652-013-0179-6.

[42] M. Hafiz, P. Adamczyk, and R. Johnson, "Growing a pattern language (for security)," *SPLASH 2012: Onward! 2012 - Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, pp. 139–158, 2012. DOI: 10.1145/2384592.2384607.

[43] J. Janulevičius, L. Marozas, A. Čenys, N. Goranin, and S. Ramanauskaite, "Enterprise architecture modeling based on cloud computing security ontology as a reference model," *2017 Open Conference of Electrical, Electronic and Information Sciences, eStream 2017 - Proceedings of the Conference*, 2017. DOI: 10.1109/eStream.2017.7950320.

[44] E. Y. Nakagawa, M. Becker, and J. C. Maldonado, "A Knowledge-Based Framework for Reference Architectures," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ser. SAC '12, New York, NY, USA: Association for Computing Machinery, 2012, pp. 1197–1202, ISBN: 9781450308571. DOI: 10.1145/2245276.2231964.

[45] O. E. Demir, P. Devanbu, N. Medvidovic, and E. Wohlstadter, "DISCOA: Architectural adaptations for security and QoS," *SESS 2005 - Proceedings of the 2005 Workshop on Software Engineering for Secure Systems - Building Trustworthy Applications*, pp. 1–7, 2005.

[46] N. Baloyi and P. Kotzé, "A data privacy model based on internet of things and cyber-physical systems reference architectures," *ACM International Conference Proceeding Series*, pp. 258–267, 2018. DOI: 10.1145/3278681.3278712.

[47] M. Pulkkinen, A. Naumenko, and K. Luostarinen, "Managing Information Security in a Business Network of Machinery Maintenance Services Business - Enterprise Architecture as a Coordination Tool," *J. Syst. Softw.*, vol. 80, no. 10, pp. 1607–1620, Oct. 2007, ISSN: 0164-1212. DOI: 10.1016/j.jss.2007.01.044.

[48] M. S. Lund, B. Solhaug, and K. Stølen, *Model-driven risk analysis: the CORAS approach*. Springer Science & Business Media, 2010.

[49] N. Alhadad, P. Serrano-Alvarado, Y. Busnel, and P. Lamarre, "System modeling and trust evaluation of distributed systems," *Lecture Notes in Computer Science and Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9430, pp. 33–74, 2015, ISSN: 16113349. DOI: 10.1007/978-3-662-48567-5_2.

[50] *Call for Papers on the special issue:Understanding and planning shared micro-mobility - Call for Papers - Elsevier*. [Online]. Available: https://www.journals.elsevier.com/transportation-research-part-d-transport-and-environment/call-for-papers/understanding%20and%20planning%20shared%20micro-mobility (visited on 05/30/2020).

[51] A. Chang, L. Miranda-Moreno, and L. S. Regina Clewlow, "Trend or fad? Deciphering the Enablers of Micromobility in the U.S.," *SAE International.*, no. July, 2019.

[52] N. Vinayaga-Sureshkanth, R. Wijewickrama, A. Maiti, and M. Jadliwala, "Security and Privacy Challenges in Upcoming Intelligent Urban Micromobility Transportation Systems," *arXiv:2001.01387*, 2020. eprint: 2001.01387.

[53] L. Li, K. Y. Lee, and S.-B. Yang, "Do Micro-Mobility Services Take Away Our Privacy? Focusing on the Privacy Paradox in E-Scooter Sharing Platforms Research-in-Progress," *Pacific Asia Conference on Information Systems*, 2019.

[54] L. Cameron Booth and M. Mayrany, "IoT Penetration Testing: Hacking an Electric Scooter Bachelor Thesis Report," *KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science*, 2019. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1334205/FULLTEXT01.pdf.

[55] M. E. Morales-Trujillo, B. Escalante-Ramírez, M. del Pilar Angeles, H. Oktaba, and G. I. González, "Towards a representation of enterprise architecture based on zachman framework through omg standards (s).," in *SEKE*, 2018, pp. 225–224.

[56] D. Garcia, "Leaking privacy and shadow profiles in online social networks," *Science Advances*, vol. 3, no. 8, 2017, ISSN: 23752548. DOI: 10.1126/sciadv.1701172.

[57] N. Baloyi and P. Kotzé, "Guidelines for data privacy compliance: A focus on cyber-physical systems and internet of things," *ACM International Conference Proceeding Series*, 2019. DOI: 10.1145/3351108.3351143.

[58] N. Baloyi and P. Kotzé, "Data privacy compliance benefits for organisations – A cyber-physical systems and internet of things study," *Communications in Computer and Information Science*, vol. 1166 CCIS, pp. 158–172, 2020, ISSN: 18650937. DOI: 10.1007/978-3-030-43276-8_12.

[59] M. Whitman, C. Y. Hsiang, and K. Roark, "Potential for participatory big data ethics and algorithm design: A scoping mapping review," *ACM International Conference Proceeding Series*, vol. 2, pp. 1–6, 2018. DOI: 10.1145/3210604.3210644.

[60] W. Xiong and R. Lagerström, "Threat modeling of connected vehicles: A privacy analysis and extension of vehiclelang," *2019 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cyber SA 2019*, 2019. DOI: 10.1109/CyberSA.2019.8899615.

[61] W. X. B, G. Melek, and K. M. Kaya, "A Study of Security Vulnerabilities," pp. 204–218, 2019. DOI: 10.1007/978-3-030-35055-0. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-35055-0_1.

[62] W. Xiong, F. Krantz, and R. Lagerström, "Threat Modeling and Attack Simulations of Connected Vehicles: Proof of Concept," *Communications in Computer and Information Science*, vol. 1221 CCIS, pp. 272–287, 2020, ISSN: 18650937. DOI: 10.1007/978-3-030-49443-8_13.

[63] R. Messnarz, G. Macher, J. Stolfa, and S. Stolfa, *Highly Autonomous Vehicle (System) Design Patterns – Achieving Fail Operational and High Level of Safety and Security*, November 2017. Springer International Publishing, 2019, vol. 1060, pp. 465–477, ISBN: 9783030280048. DOI: 10.1007/978-3-030-28005-5_36. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-28005-5_36.

[64] M. S. Chong, H. Sandberg, and A. M. Teixeira, "A tutorial introduction to security and privacy for cyber-physical systems," *2019 18th European Control Conference, ECC 2019*, pp. 968–978, 2019. DOI: 10.23919/ECC.2019.8795652.

[65] M. Mahbub, "Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics," *Journal of Network and Computer Applications*, vol. 168, no. April, p. 102 761, 2020, ISSN: 10958592. DOI: 10.1016/j.jnca.2020.102761. [Online]. Available: https://doi.org/10.1016/j.jnca.2020.102761.

[66] A. A. Ganin, P. Quach, M. Panwar, Z. A. Collier, J. M. Keisler, D. Marchese, and I. Linkov, "Multicriteria Decision Framework for Cybersecurity Risk Assessment and Management," *Risk Analysis*, vol. 40, no. 1, pp. 183–199, 2020, ISSN: 15396924. DOI: 10.1111/risa.12891.

[67] J. Cui, G. Sabaliauskaite, L. S. Liew, F. Zhou, and B. Zhang, "Collaborative Analysis Framework of Safety and Security for Autonomous Vehicles," *IEEE Access*, vol. 7, no. March 2018, pp. 148 672–148 683, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2946632.

[68] A. M. Shaaban, A. Baith Mohamed, C. Schmittner, G. Quirchmayr, T. Gruber, and E. Schikuta, "CloudWoT - A reference model for knowledge-based IoT solutions," *ACM International Conference Proceeding Series*, pp. 272–281, 2018. DOI: 10.1145/3282373.3282400.

[69] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4118–4149, 2019, ISSN: 23274662. DOI: 10.1109/JIOT.2018.2875544.

[70] S. Ouchani and A. Khaled, "A meta language for cyber-physical systems and threats: Application on autonomous vehicle," *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, vol. 2019-Novem, 2019, ISSN: 21615330. DOI: 10.1109/AICCSA47632.2019.9035273.

[71] T. Kulik, P. W. Tran-Jørgensen, J. Boudjadar, and C. Schultz, "A framework for threat-driven cyber security verification of IoT Systems," *Proceedings - 2018 IEEE 11th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2018*, pp. 89–97, 2018. DOI: 10.1109/ICSTW.2018.00033.

[72] J. Augusto-Gonzalez, A. Collen, S. Evangelatos, M. Anagnostopoulos, G. Spathoulas, K. M. Giannoutakis, K. Votis, D. Tzovaras, B. Genge, E. Gelenbe, and N. A. Nijdam, "From internet of threats to internet of things: A cyber security architecture for smart homes," *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, vol. 2019-Septe, 2019, ISSN: 23784873. DOI: 10.1109/CAMAD.2019.8858493.

[73] J. Janulevičius and L. Šiaudinytė, "Enterprise Architecture Analysis for Security Issue Diagnositics in Distributed Systems," pp. 1–4, 2015.

[74] G. Kavallieratos, S. Katsikas, and V. Gkioulos, "SafeSec Tropos: Joint security and safety requirements elicitation," *Computer Standards and Interfaces*, vol. 70, no. January, p. 103 429, 2020, ISSN: 09205489. DOI: 10.1016/j.csi.2020.103429. [Online]. Available: https://doi.org/10.1016/j.csi.2020.103429.

[75] S. Katsikeas, P. Johnson, S. Hacks, and R. Lagerström, "Probabilistic modeling and simulation of vehicular cyber attacks: An application of the meta attack language," *ICISSP 2019 - Proceedings of the 5th International Conference on Information Systems Security and Privacy*, pp. 175–182, 2019. DOI: 10.5220/0007247901750182.

[76] M. Koschuch, W. Sebron, Z. Szalay, A. Torok, H. Tschiurtz, and I. Wahl, "Safety & security in the context of autonomous driving," *2019 8th IEEE International Conference on Connected Vehicles and Expo, ICCVE 2019 - Proceedings*, 2019. DOI: 10.1109/ICCVE45908.2019.8965092.

[77] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," *ACM International Conference Proceeding Series*, 2018. DOI: 10.1145/3230833.3232799.

[78] A. Zimmermann, R. Schmidt, K. Sandkuhl, M. Wißotzki, D. Jugel, and M. Möhring, "Digital enterprise architecture-transformation for the internet of things," *Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOCW 2015*, no. September, pp. 130–138, 2015. DOI: 10.1109/EDOCW.2015.16.

[79] O. M. Latvala, J. Toivonen, J. Kuusijärvi, and A. Evesti, "A tool for security metrics modeling and visualization," *ACM International Conference Proceeding Series*, 2014. DOI: 10.1145/2642803.2642806.

[80] A. L. Zhang, "Research on the architecture of internet of things applied in coal mine," *Proceedings - 2016 International Conference on Information System and Artificial Intelligence, ISAI 2016*, pp. 21–23, 2017. DOI: 10.1109/ISAI.2016.0014.

[81] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (SIoT) - When social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, 2012, ISSN: 13891286. DOI: 10.1016/j.comnet.2012.07.010.

[82] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, 2016, ISBN: 9781908320520. DOI: 10.1109/ICITST.2015.7412116.

[83] M. Leo, F. Battisti, M. Carli, and A. Neri, "A federated architecture approach for Internet of Things security," in *2014 Euro Med Telco Conference - From Network Infrastructures to Network Fabric: Revolution at the Edges, EMTC 2014*, 2014, ISBN: 9788887237207. DOI: 10.1109/EMTC.2014.6996632.

[84] F. Bing, "The research of IOT of agriculture based on three layers architecture," in *Proceedings of 2016 2nd International Conference on Cloud Computing and Internet of Things, CCIOT 2016*, 2017, ISBN: 9781467398213. DOI: 10.1109/CCIOT.2016.7868325.

[85] N. Kaur and S. K. Sood, "An Energy-Efficient Architecture for the Internet of Things (IoT)," vol. 11, no. 2, pp. 796–805, 2017.

[86] A. Torkaman and M. A. Seyyedi, "Analyzing IoT Reference Architecture Models," *International Journal of Computer Science and Software Engineering ISSN*, vol. 5, no. 8, pp. 2409–4285, 2016. [Online]. Available: `www.IJCSSE.org`.

[87] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, 2017, ISSN: 23274662. DOI: `10.1109/JIOT.2017.2683200`.

[88] Industrial Internet Consortium, "Industrial Internet Reference Architecture," *Technical Report*, 2015, ISSN: 1932-6203.

[89] A. Morkevicius, L. Bisikirskiene, and G. Bleakley, "Using a systems of systems modeling approach for developing Industrial Internet of Things applications," *2017 12th System of Systems Engineering Conference, SoSE 2017*, no. June, 2017. DOI: `10.1109/SYSOSE.2017.7994942`.

[90] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," *Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012*, pp. 257–260, 2012. DOI: `10.1109/FIT.2012.53`.

[91] H. Freeman and T. Zhang, "The emerging era of fog computing and networking [The President's Page]," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 4–5, 2016, ISSN: 01636804. DOI: `10.1109/MCOM.2016.7497757`.

[92] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. S. D. Souza, and V. Trifa, "Soa-based integration of the internet of things in enterprise services," in *2009 IEEE International Conference on Web Services, ICWS 2009*, 2009, ISBN: 9780769537092. DOI: `10.1109/ICWS.2009.98`.

[93] L. H. B. Tietz, "Development of an Architecture for a Tele-Medicine-Based Longterm Monitoring System," no. April, 2016.

[94] A. Varghese and D. Tandur, "Wireless requirements and challenges in Industry 4.0," *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, pp. 634–638, 2014. DOI: `10.1109/IC3I.2014.7019732`.

[95] M. Keller, M. Rosenberg, M. Brettel, and N. Friederichsen, "How Virtualization, Decentrazliation and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective," *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, vol. 8, no. 1, pp. 37–44, 2014.

[96] M. Broy, "Cyber-Physical Systems Innovation durch Software-Intensive eingebettet Systeme," *acatech Diskutiert*, pp. 1–141, 2010, ISSN: 1098-6596. arXiv: arXiv:1011.1669v3. [Online]. Available: http://www.acatech.de/de/publikationen/berichte-und-dokumentationen/acatech/detail/artikel/cyber-physical-systems-innovation-durch-softwareintensive-eingebetete-systeme.html%7B%5C%%7D5Cnhttp://link.springer.com/10.1007/978-3-642-14901-6.

[97] J. Schlechtendahl, M. Keinert, F. Kretschmer, A. Lechler, and A. Verl, "Making existing production systems Industry 4.0-ready: Holistic approach to the integration of existing production systems in Industry 4.0 environments," *Production Engineering*, vol. 9, no. 1, pp. 143–148, 2014, ISSN: 18637353. DOI: 10.1007/s11740-014-0586-3.

[98] L. Shi-Wan, M. Bradford, D. Jacques, B. Graham, A. Chigani, R. Martin, B. Murphy, and M. Crawford, "The Industrial Internet of Things Volume G1 : Reference Architecture," *Industrial Internet Consortium White Paper*, 2017.

[99] J. Lee, B. Bagheri, and H. A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manufacturing Letters*, 2015, ISSN: 22138463. DOI: 10.1016/j.mfglet.2014.12.001.

[100] S. V. Thieven, A. Cliton, M. Mahto, and B. Sniderman, "chemicals industry Catalyzing transformation," *Deloitte Univeristy Press*, pp. 1–24, 2016.

[101] VID/VDE, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," *Igarss 2014*, 2015, ISSN: 0717-6163. arXiv: arXiv:1011.1669v3.

[102] D. Navani, S. Jain, and M. S. Nehra, "The internet of things (IoT): A study of architectural elements," *Proceedings - 13th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2017*, vol. 2018-Janua, pp. 473–478, 2018. DOI: 10.1109/SITIS.2017.83.

[103] M. S. Virat, S. M. Bindu, B. Aishwarya, B. N. Dhanush, and M. R. Kounte, "Security and Privacy Challenges in Internet of Things," *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, no. Icoei, pp. 454–460, 2018. DOI: 10.1109/ICOEI.2018.8553919.

[104] P. Patel and D. Cassou, "Enabling high-level application development for the Internet of Things," in *Journal of Systems and Software*, 2015. DOI: 10.1016/j.jss.2015.01.027. arXiv: 1501.05080.

[105] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, "Research on the architecture of Internet of Things," *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, vol. 5, no. September 2010, 2010. DOI: 10.1109/ICACTE.2010.5579493.

[106] M. Weyrich and C. Ebert, "Reference architectures for the internet of things," *IEEE Software*, 2016, ISSN: 07407459. DOI: 10.1109/MS.2016.20.

[107] G. Kavallieratos, S. Katsikas, and V. Gkioulos, "Cybersecurity and safety co-engineering of cyberphysical systems - A comprehensive survey," *Future Internet*, vol. 12, no. 4, pp. 1–17, 2020, ISSN: 19995903. DOI: 10.3390/FI12040065.

[108] N. Boucké and T. Holvoet, "Relating architectural views with architectural concerns," *Proceedings - International Conference on Software Engineering*, pp. 11–17, 2006, ISSN: 02705257. DOI: 10.1145/1137639.1137643.

[109] H. Mahdikhani, S. Mahdavifar, R. Lu, H. Zhu, and A. A. Ghorbani, "Achieving privacy-preserving subset aggregation in fog-enhanced IoT," *IEEE Access*, vol. 7, pp. 184 438–184 447, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2961270.

[110] Y. Ming and X. Yu, "Efficient privacy-preserving data sharing for fog-assisted vehicular sensor networks," *Sensors (Switzerland)*, vol. 20, no. 2, 2020, ISSN: 14248220. DOI: 10.3390/s20020514.

[111] G. Sabaliauskaite, J. Cui, and L. S. Liew, "Integrating Autonomous Vehicle Safety and Security Analysis Using STPA Method and the Six-Step Model Autonomous Vehicle Security View project Autonomous Vehicle Security View project Integrating Autonomous Vehicle Safety and Security Analysis Using STPA M," no. July, 2018. [Online]. Available: https://www.researchgate.net/publication/326504334.

[112] G. Sabaliauskaite and A. P. Mathur, "Aligning Cyber-Physical System Safety and Security," in *Complex Systems Design & Management Asia*, 2015. DOI: 10.1007/978-3-319-12544-2_4.

[113] J. Cui and G. Sabaliauskaite, "On the Alignment of Safety and Security for Autonomous Vehicles," no. November 2017, pp. 59–64, 2018.

[114] F. Asplund, J. McDermid, R. Oates, and J. Roberts, "Rapid Integration of CPS Security and Safety," *IEEE Embedded Systems Letters*, 2019, ISSN: 19430671. DOI: 10.1109/LES.2018.2879631.

[115] N. H. Carreras Guzman, D. Kwame Minde Kufoalor, I. Kozine, and M. A. Lundteigen, "Combined safety and security risk analysis using the UFoI-E method: A case study of an autonomous surface vessel," in *Proceedings of the 29th European Safety and Reliability Conference, ESREL 2019*, 2020, ISBN: 9789811127243. DOI: 10.3850/978-981-11-2724-3_0208-cd.

[116] G. Sabaliauskaite, L. S. Liew, and F. Zhou, "AVES – Automated vehicle safety and security analysis framework," in *Proceedings - CSCS 2019: ACM Computer Science in Cars Symposium*, 2019, ISBN: 9781450370042. DOI: 10.1145/3359999.3360494.

[117] N. H. Carreras Guzman, M. Wied, I. Kozine, and M. A. Lundteigen, "Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis," *Systems Engineering*, 2020, ISSN: 15206858. DOI: 10.1002/sys.21509.

[118] T. Hayakawa, R. Sasaki, H. Hayashi, Y. Takahashi, T. Kaneko, and T. Okubo, "Proposal and Application of Security/Safety Evaluation Method for Medical Device System That Includes IoT," in *Proceedings of the 2018 VII International Conference on Network, Communication and Computing*, ser. ICNCC 2018, New York, NY, USA: Association for Computing Machinery, 2018, pp. 157–164, ISBN: 9781450365536. DOI: 10.1145/3301326.3301330. [Online]. Available: https://doi.org/10.1145/3301326.3301330.

[119] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "SAHARA: A security-aware hazard and risk analysis method," in *Proceedings -Design, Automation and Test in Europe, DATE*, 2015, ISBN: 9783981537048. DOI: 10.7873/date.2015.0622.

[120] A. Seeam, O. S. Ogbeh, S. Guness, and X. Bellekens, "Threat Modeling and Security Issues for the Internet of Things," *2nd International Conference on Next Generation Computing Applications 2019, NextComp 2019 - Proceedings*, pp. 1–8, 2019. DOI: 10.1109/NEXTCOMP.2019.8883642.

[121] M. H. Bhuyan, N. A. Azad, W. Meng, and C. D. Jensen, "Analyzing the communication security between smartphones and IoT based on CORAS," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, ISBN: 9783030027438. DOI: 10.1007/978-3-030-02744-5_19.

[122] B. L. Papke, "Enabling design of agile security in the IOT with MBSE," *2017 12th System of Systems Engineering Conference, SoSE 2017*, pp. 1–6, 2017. DOI: 10.1109/SYSOSE.2017.7994938.

[123] Y. Park, S. Daftari, P. Inamdar, S. Salavi, A. Savanand, and Y. Kim, "IoTGuard: Scalable and agile safeguards for Internet of Things," in *Proceedings - IEEE Military Communications Conference MILCOM*, 2016, ISBN: 9781509037810. DOI: 10.1109/MILCOM.2016.7795302.

[124] J. Pacheco and S. Hariri, "IoT security framework for smart cyber infrastructures," in *Proceedings - IEEE 1st International Workshops on Foundations and Applications of Self-Systems, FAS-W 2016*, 2016, ISBN: 9781509036516. DOI: 10.1109/FAS-W.2016.58.

[125] L. Rafferty, F. Iqbal, S. Aleem, Z. Lu, S. C. Huang, and P. C. Hung, "Intelligent multi-agent collaboration model for smart home IoT security," in *Proceedings - 2018 IEEE International Congress on Internet of Things, ICIOT 2018 - Part of the 2018 IEEE World Congress on Services*, 2018, ISBN: 9781538672440. DOI: 10.1109/ICIOT.2018.00016.

[126] J. Habibi, D. Midi, A. Mudgerikar, and E. Bertino, "Heimdall: Mitigating the Internet of Insecure Things," *IEEE Internet of Things Journal*, 2017, ISSN: 23274662. DOI: 10.1109/JIOT.2017.2704093.

[127] N. DeMarinis and R. Fonseca, "Toward usable network traffic policies for iot devices in consumer networks," in *IoT S and P 2017 - Proceedings of the 2017 Workshop on Internet of Things Security and Privacy, co-located with CCS 2017*, 2017, ISBN: 9781450353960. DOI: 10.1145/3139937.3139949.

[128] M. Serror, M. Henze, S. Hack, M. Schuba, and K. Wehrle, "Towards in-network security for smart homes," in *ACM International Conference Proceeding Series*, 2018, ISBN: 9781450364485. DOI: 10.1145/3230833.3232802.

[129] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 2017, ISBN: 9781509043385. DOI: 10.1109/PERCOMW.2017.7917634.

[130] S. Li, F. Bi, W. Chen, X. Miao, J. Liu, and C. Tang, "An improved information security risk assessments method for cyber-physical-social computing and networking," *IEEE Access*, vol. 6, pp. 10 311–10 319, 2018, ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2800664.

[131] A. Riel, C. Kreiner, R. Messnarz, and A. Much, "An architectural approach to the integration of safety and security requirements in smart products and systems design," *CIRP Annals*, vol. 67, no. 1, pp. 173–176, 2018, ISSN: 17260604. DOI: 10.1016/j.cirp.2018.04.022. [Online]. Available: https://doi.org/10.1016/j.cirp.2018.04.022.

[132] C. Islam, M. A. Babar, and S. Nepal, "A multi-vocal review of security orchestration," *ACM Computing Surveys*, vol. 52, no. 2, 2019, ISSN: 15577341. DOI: 10.1145/3305268.

[133] B. Yigit Ozkan, M. Spruit, R. Wondolleck, and V. Burriel Coll, "Modelling adaptive information security for SMEs in a cluster," *Journal of Intellectual Capital*, 2019, ISSN: 17587468. DOI: 10.1108/JIC-05-2019-0128.

[134] M. Spruit and M. Roeling, "ISFAM: The information security focus area maturity model," in *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, 2014, ISBN: 9780991556700.

[135] U.S. Department of Energy, "Electricity subsector cybersecurity capability maturity model," no. February, p. 89, 2014. [Online]. Available: http://energy.gov/sites/prod/files/2014/02/f7/ES-C2M2-v1-1-Feb2014.pdf.

[136] The Open Group, "Open Group Standard Open Information Security Management Maturity Model," *ISM3 Consortium*, 2011.

[137] US Department of Homeland Security, "Department of Homeland Security Cybersecurity Capability Maturity Model White Paper," 2014.

[138] V. Hernández, L. López, O. Prieto, J. F. Martínez, A. B. García, and A. Da-Silva, "Security framework for DPWS compliant devices," *Proceedings - 2009 3rd International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009*, pp. 87–92, 2009. DOI: 10.1109/SECURWARE.2009.21.

[139] ISA/IEC, *ISA/IEC 62443 series*, 2018. [Online]. Available: https://www.isa.org/intech-home/2018/september-october/departments/new-standard-specifies-security-capabilities-for-c (visited on 11/16/2020).

[140] ISO/SAE, *ISO - ISO/SAE DIS 21434 - Road vehicles — Cybersecurity engineering*, 2020. [Online]. Available: https://www.iso.org/standard/70918.html (visited on 11/16/2020).

[141] ISO/IEC, *ISO - ISO/IEC 27000:2018 - Information technology — Security techniques — Information security management systems — Overview and vocabulary*, 2018. [Online]. Available: https://www.iso.org/standard/73906.html (visited on 11/16/2020).

[142] ISO/IEC, *ISO - ISO/IEC 27034-1:2011 - Information technology — Security techniques — Application security — Part 1: Overview and concepts*, 2011. [Online]. Available: https://www.iso.org/standard/44378.html?browse=tc (visited on 11/16/2020).

[143] ISO/IEC, *ISO - ISO/IEC 29100:2011 - Information technology — Security techniques — Privacy framework*, 2011. [Online]. Available: https://www.iso.org/standard/45123.html (visited on 11/16/2020).

[144] NIST, *NIST SP 800-30 — NIST*, 2020. [Online]. Available: https://www.nist.gov/privacy-framework/nist-sp-800-30 (visited on 11/16/2020).

[145] E. J. McGuire, "Safety and security," *Project Management: A Reference for Professionals*, pp. 537–540, 2017. DOI: 10.1201/9780203741771.

[146] NIST, *NVD - 800-53*. [Online]. Available: https://nvd.nist.gov/800-53 (visited on 11/16/2020).

[147] SAE, *J3061A (WIP) Cybersecurity Guidebook for Cyber-Physical Vehicle Systems - SAE International*, 2016. [Online]. Available: https://www.sae.org/standards/content/j3061/ (visited on 11/16/2020).

[148] J. Cui and G. Sabaliauskaite, "US 2 : An unified safety and security analysis method for autonomous vehicles," in *Advances in Intelligent Systems and Computing*, 2019, ISBN: 9783030034016. DOI: 10.1007/978-3-030-03402-3_42.

[149] ISO, *ISO - ISO 31000:2018 - Risk management — Guidelines*, 2018. [Online]. Available: https://www.iso.org/standard/65694.html (visited on 11/16/2020).

[150] IEC, *IEC 62645:2019 — IEC Webstore*, 2019. [Online]. Available: https://webstore.iec.ch/publication/32904 (visited on 11/16/2020).

[151] IEC, *IEC 62859:2016 — IEC Webstore — cyber security, energy*, 2016. [Online]. Available: https://webstore.iec.ch/publication/26131 (visited on 11/16/2020).

[152] CENELEC, *EN 50126 : 1999 — RAILWAY APPLICATIONS - THE SPECIFICATION AND DEMONSTRATION OF RELIABILITY, AVAILABILITY, MAINTAINABILITY AND SAFETY (RAMS) — SAI Global*, 1999. [Online]. Available: https://infostore.saiglobal.com/en-us/Standards/EN-50126-1999-353881_SAIG_CENELEC_CENELEC_807075/ (visited on 11/16/2020).

[153] BSi, *BS EN 50128 : 2011 — RAILWAY APPLICATIONS - COMMUNICATIONS, SIGNALLING AND PROCESSING SYSTEMS - SOFTWARE FOR RAILWAY CONTROL AND PROTECTION SYSTEMS — SAI Global*, 2011. [Online]. Available: https://infostore.saiglobal.com/en-us/Standards/BS-EN-50128-2011-233224_SAIG_BSI_BSI_545882/ (visited on 11/16/2020).

[154] CENELEC, *CENELEC - EN 50129 - Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling — Engineering360*, 2018. [Online]. Available: https://standards.globalspec.com/std/13113133/EN%2050129 (visited on 11/16/2020).

[155] IEC, *IEC Functional Safety and IEC 61508*. [Online]. Available: https://www.iec.ch/functionalsafety/ (visited on 11/16/2020).

[156] ISO/PAS, *ISO - ISO/PAS 21448:2019 - Road vehicles — Safety of the intended functionality*, 2019. [Online]. Available: `https://www.iso.org/standard/70939.html` (visited on 11/16/2020).

[157] UK-MOD, *DEFSTAN 00-56(PT1)/7(2017) : 2017 — SAFETY MANAGEMENT REQUIREMENTS FOR DEFENCE SYSTEMS - PART 1: REQUIREMENTS — SAI Global*, 2017. [Online]. Available: `https://infostore.saiglobal.com/en-us/Standards/DEFSTAN-00-56-PT1-7-2017-2017-369666_SAIG_DEFSTAN_DEFSTAN_842128/` (visited on 11/16/2020).

[158] ISO, *ISO - ISO 26262-1:2018 - Road vehicles — Functional safety — Part 1: Vocabulary*, 2018. [Online]. Available: `https://www.iso.org/standard/68383.html` (visited on 11/16/2020).

[159] US-DoD, *MIL STD 882 : E — STANDARD PRACTICE FOR SYSTEM SAFETY — SAI Global*. [Online]. Available: `https://infostore.saiglobal.com/en-us/Standards/MIL-STD-882-E-733701_SAIG_MIL_MIL_1702968/` (visited on 11/16/2020).

[160] IEC, *IEC 61499-1:2005 — IEC Webstore*, 2005. [Online]. Available: `https://webstore.iec.ch/publication/19794` (visited on 11/16/2020).

[161] ISO/IEC/IEEE, *ISO - ISO/IEC/IEEE 42010:2011 - Systems and software engineering — Architecture description*, 2011. [Online]. Available: `https://www.iso.org/standard/50508.html` (visited on 11/16/2020).

[162] ECSS, *ECSS-Q-ST-80C Rev.1 – Software product assurance (15 February 2017) — European Cooperation for Space Standardization*, 2017. [Online]. Available: `https://ecss.nl/standard/ecss-q-st-80c-rev-1-software-product-assurance-15-february-2017/` (visited on 11/16/2020).

[163] ISO/IEC, *ISO - ISO/IEC 25010:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*, 2011. [Online]. Available: `https://www.iso.org/standard/35733.html` (visited on 11/16/2020).

[164] SAE, *J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems - SAE International*, 2014. [Online]. Available: `https://www.sae.org/standards/content/j3016_201401/` (visited on 11/16/2020).

[165] A. R. Berkel, P. M. Singh, and M. J. van Sinderen, "An Information Security Architecture for Smart Cities," *Lecture Notes in Business Information Processing*, vol. 319, no. September, pp. 167–184, 2018, ISSN: 18651348. DOI: `10.1007/978-3-319-94214-8_11`.

[166] S. Hacks, A. Hacks, S. Katsikeas, B. Klaer, and R. Lagerstrom, "Creating meta attack language instances using archimate: Applied to electric power and energy system cases," in *Proceedings - 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference, EDOC 2019*, 2019, ISBN: 9781728127026. DOI: 10.1109/EDOC.2019.00020.

[167] P. Närman, U. Franke, J. König, M. Buschle, and M. Ekstedt, "Enterprise architecture availability analysis using fault trees and stakeholder interviews," *Enterprise Information Systems*, vol. 8, no. 1, pp. 1–25, 2014, ISSN: 17517575. DOI: 10.1080/17517575.2011.647092.

[168] Q. Zhi, S. Yamamoto, and S. Morisaki, "Quantitative Evaluation in Security Assurance," *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 2477–2483, 2019. DOI: 10.1109/compcomm.2018.8780877.

[169] Q. Zhi, S. Yamamoto, and S. Morisaki, "IMSA - Intra Model Security Assurance," vol. 2, no. May, pp. 18–32, 2018.

[170] W. Xiong, P. Carlsson, and R. Lagerström, "Re-using enterprise architecture repositories for agile threat modeling," *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, vol. 2019-Octob, pp. 118–127, 2019, ISSN: 15417719. DOI: 10.1109/EDOCW.2019.00031.

[171] V. Nagaraju, L. Fiondella, and T. Wandji, "A survey of fault and attack tree modeling and analysis for cyber risk management," *2017 IEEE International Symposium on Technologies for Homeland Security, HST 2017*, 2017. DOI: 10.1109/THS.2017.7943455.

[172] A. E. M. Al-Dahasi and B. N. A. Saqib, "Attack tree Model for Potential Attacks Against the SCADA System," *27th Telecommunications Forum, TELFOR 2019*, pp. 9–12, 2019. DOI: 10.1109/TELFOR48224.2019.8971181.

[173] A. Buldas, O. Gadyatskaya, A. Lenin, S. Mauw, and R. Trujillo-Rasua, "Attribute evaluation on attack trees with incomplete information," *Computers and Security*, vol. 88, p. 101 630, 2020, ISSN: 01674048. DOI: 10.1016/j.cose.2019.101630. [Online]. Available: https://doi.org/10.1016/j.cose.2019.101630.

[174] W. Wideł, M. Audinot, B. Fila, and S. Pinchinat, "Beyond 2014: Formal methods for attack tree-based security modeling," *ACM Computing Surveys*, vol. 52, no. 4, 2019, ISSN: 15577341. DOI: 10.1145/3331524.

[175] B. Kordy, M. Pouly, and P. Schweitzer, "Probabilistic reasoning with graphical security models," *Information Sciences*, vol. 342, pp. 111–131, 2016, ISSN: 00200255. DOI: 10.1016/j.ins.2016.01.010. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2016.01.010.

[176] X. Mao, M. Ekstedt, E. Ling, E. Ringdahl, and R. Lagerström, "Conceptual Abstraction of Attack Graphs - A Use Case of securiCAD," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11720 LNCS, pp. 186–202, 2019, ISSN: 16113349. DOI: 10.1007/978-3-030-36537-0_9.

[177] S. Zhang, X. Ou, and J. Homer, "through Model Abstraction," pp. 17–34, 2011.

[178] M. H. Khouzani, Z. Liu, and P. Malacaria, "Scalable min-max multi-objective cyber-security optimisation over probabilistic attack graphs," *European Journal of Operational Research*, vol. 278, no. 3, pp. 894–903, 2019, ISSN: 03772217. DOI: 10.1016/j.ejor.2019.04.035. [Online]. Available: https://doi.org/10.1016/j.ejor.2019.04.035.

[179] T. Sommestad, M. Ekstedt, and P. Johnson, "A probabilistic relational model for security risk analysis," *Computers and Security*, vol. 29, no. 6, pp. 659–679, 2010, ISSN: 01674048. DOI: 10.1016/j.cose.2010.02.002. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2010.02.002.

[180] E. Lamine, R. Thabet, A. Sienou, D. Bork, F. Fontanili, and H. Pingaud, "BPRIM: An integrated framework for business process management and risk management," *Computers in Industry*, vol. 117, p. 103 199, 2020, ISSN: 01663615. DOI: 10.1016/j.compind.2020.103199. [Online]. Available: https://doi.org/10.1016/j.compind.2020.103199.

[181] O. M. Latvala, J. Toivonen, A. Evesti, M. Sihvonen, and V. Jordan, "Security Risk Visualization with Semantic Risk Model," *Procedia Computer Science*, vol. 83, pp. 1194–1199, 2016, ISSN: 18770509. DOI: 10.1016/j.procs.2016.04.247. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2016.04.247.

[182] M. Buschle, H. Holm, T. Sommestad, M. Ekstedt, and K. Shahzad, "A tool for automatic enterprise architecture modeling," *Lecture Notes in Business Information Processing*, vol. 107 LNBIP, pp. 1–15, 2012, ISSN: 18651348. DOI: 10.1007/978-3-642-29749-6_1.

[183] M. Buschle, J. Ullberg, U. Franke, R. Lagerström, and T. Sommestad, "A tool for enterprise architecture analysis using the PRM formalism," *CEUR Workshop Proceedings*, vol. 592, pp. 108–121, 2010, ISSN: 16130073.

[184] R. Lagerström, P. Johnson, and M. Ekstedt, "Automatic design of secure enterprise architecture: Work in progress paper," *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, vol. 2017-Octob, pp. 65–70, 2017, ISSN: 15417719. DOI: 10.1109/EDOCW.2017.19.

[185] D. Falessi, S. Nejati, M. Sabetzadeh, L. Briand, and A. Messina, "SafeSlice: A Model Slicing and Design Safety Inspection Tool for SysML," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ser. ESEC/FSE '11, New York, NY, USA: Association for Computing Machinery, 2011, pp. 460–463, ISBN: 9781450304436. DOI: 10.1145/2025113.2025191. [Online]. Available: https://doi.org/10.1145/2025113.2025191.

[186] S. Nejati, M. Sabetzadeh, D. Falessi, L. Briand, and T. Coq, "A SysML-based approach to traceability management and design slicing in support of safety certification: Framework, tool support, and case studies," *Information and Software Technology*, vol. 54, no. 6, pp. 569–590, 2012, ISSN: 09505849. DOI: 10.1016/j.infsof.2012.01.005. [Online]. Available: http://dx.doi.org/10.1016/j.infsof.2012.01.005.

[187] C. Easttom, "SecML: A Proposed Modeling Language for CyberSecurity," *2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019*, pp. 1015–1021, 2019. DOI: 10.1109/UEMCON47517.2019.8993105.

[188] M. Ekstedt, P. Johnson, R. Lagerström, D. Gorton, J. Nydrén, and K. Shahzad, "SecuriCAD by foreseeti: A CAD tool for enterprise cyber security management," *Proceedings of the 2015 IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, EDOCW 2015*, pp. 152–155, 2015. DOI: 10.1109/EDOCW.2015.40.

[189] T. Sommestad, M. Ekstedt, and H. Holm, "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," *IEEE Systems Journal*, vol. 7, no. 3, pp. 363–373, 2013, ISSN: 19328184. DOI: 10.1109/JSYST.2012.2221853.

[190] H. Holm, K. Shahzad, M. Buschle, and M. Ekstedt, "P2CySeMoL: Predictive, Probabilistic Cyber Security Modeling Language," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 626–639, 2015, ISSN: 19410018. DOI: 10.1109/TDSC.2014.2382574.

[191] E. Wohlstadter, S. Jackson, and P. Devanbu, "DADO: Enhancing middleware to support crosscutting features in distributed, heterogeneous systems," in *Proceedings - International Conference on Software Engineering*, 2003. DOI: 10.1109/icse.2003.1201198.

[192] Z. Zhou, Q. Zhi, S. Morisaki, and S. Yamamoto, "A Systematic Literature Review on Enterprise Architecture Visualization Methodologies," *IEEE Access*, vol. 8, pp. 96 404–96 427, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2995850.

[193] D. Mažeika and R. Butleris, "Integrating Security Requirements Engineering into MBSE: Profile and Guidelines," *Security and Communication Networks*, vol. 2020, no. i, 2020, ISSN: 19390122. DOI: 10.1155/2020/5137625.

[194] W. Xiong and R. Lagerström, "Threat modeling – A systematic literature review," *Computers and Security*, vol. 84, pp. 53–69, 2019, ISSN: 01674048. DOI: 10.1016/j.cose.2019.03.010. [Online]. Available: https://doi.org/10.1016/j.cose.2019.03.010.

[195] P. L. C. T. Santos, P. A. A. Monteiro, M. Studic, and A. Majumdar, *A methodology used for the development of an Air Traffic Management functional system architecture*, 2017. DOI: 10.1016/j.ress.2017.05.022.

[196] S. Chung, S. Moon, and B. Endicott-Popovsky, "Architecture-Driven Penetration Testing against an Identity Access Management (IAM) System," in *Proceedings of the 5th Annual Conference on Research in Information Technology*, ser. RIIT '16, New York, NY, USA: Association for Computing Machinery, 2016, pp. 13–18, ISBN: 9781450344531. DOI: 10.1145/2978178.2978183. [Online]. Available: https://doi.org/10.1145/2978178.2978183.

[197] S. Chung, B. Endicott-Popovsky, C. Crompton, S. H. Baeg, Y. Bai, and S. Park, "Analyses of evolving legacy software into secure service-oriented software using scrum and a visual model," in *Software Design and Development: Concepts, Methodologies, Tools, and Applications*, 2013, ISBN: 9781466643024. DOI: 10.4018/978-1-4666-4301-7.ch084.

[198] P. B. Kruchten, "The 4+1 View Model of Architecture," *IEEE Software*, 1995, ISSN: 07407459. DOI: 10.1109/52.469759.

[199] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1999, ISBN: 3540663479. DOI: 10.1007/3-540-48405-1_36.

[200] C. Dwork, "Differential privacy," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4052 LNCS, pp. 1–12, 2006, ISSN: 16113349. DOI: 10.1007/11787006_1.

[201] CBS, *Uber Crunches User Data To Determine Where The Most 'One-Night Stands' Come From – CBS San Francisco*, 2014. [Online]. Available: https://sanfrancisco.cbslocal.com/2014/11/18/uber-crunches-user-data-to-determine-where-the-most-one-night-stands-come-from/ (visited on 02/24/2021).

Appendix A

Interview Structure

Security in Micromobility

Question one:

- What are the primary security concerns that micromobility enterprise need to tackle?

Question Two:

- Micromobility services are primarily enabled through smart devices and applications. What security considerations are relevant in this domain?

Question Three:

- One important way that micromobility enterprise differs from other businesses is that they provide devices that are placed in the public domain. How does this aspect effect security considerations?

Question Four:

- What processes are used when deciding on security features?

Enterprise Architecture Modelling

Question Six:

- Do you use visual enterprise architecture modelling?
  - What languages are used in your models?
    - What primary benefits are you looking for when deciding on the modelling language?
  - Do these models involve domain layers?
    - What and how many layers constitute these models?
  - What are the motivations/benefits if these models? Who are the stakeholders?

  - Do you use any other documentation to describe the enterprises structure?

Security integration into EA modelling

Question Seven:

- Would you benefit from integrating security into your EA documentation?
  - Why?

Appendix B

Prevalent Authors

Table B.1: Security and Safety objectives defined by [74]

Begin of Table	
Author	**Publications**
Ekstedt, Mathias	[179] [182] [184] [176] [167] [35] [190] [188] [189]
Lagerström, Robert	[183] [184] [176] [75] [170] [188] [194] [62] [60]
Sommestad, Teodor	[179] [182] [183] [35] [189]

Continuation of Table 5.8	
Xiong, Wenjun	[61] [170] [194] [62] [60]
Buschle, Markus	[182] [183] [167] [190]
Johmson, Pontus	[179] [184] [75] [188]
Holm, Hannes	[182] [190] [189]
Shahzad, Khurram	[182] [190] [188]
Baloyi, Ntsako	[46] [57] [58]
Kotzé, Paula	[46] [58] [57]
End of Table	

Appendix C

Nvivo Classifications

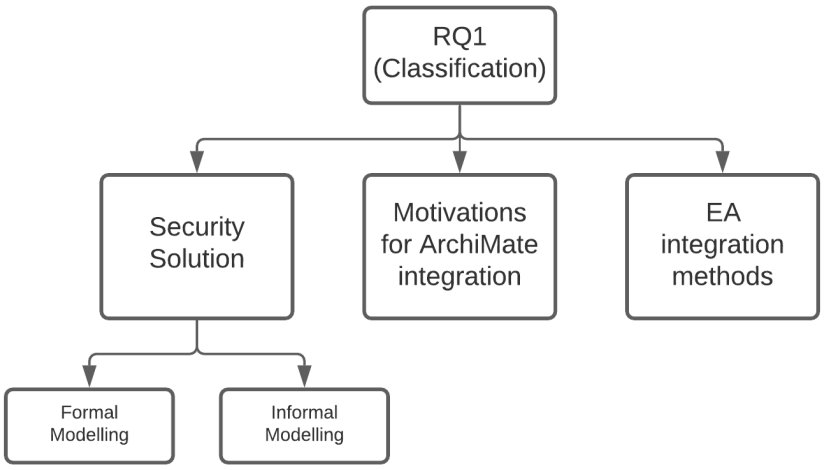


Figure C.1: Nvivo classifications for RQ1

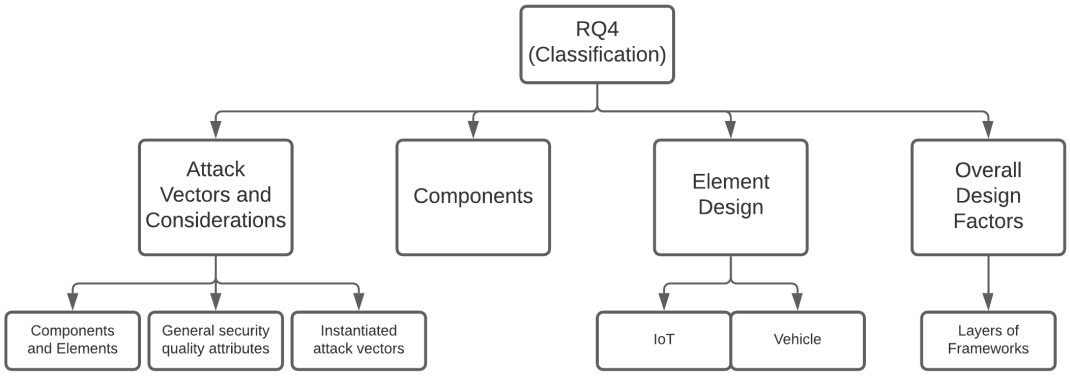


Figure C.2: Nvivo classifications for RQ2

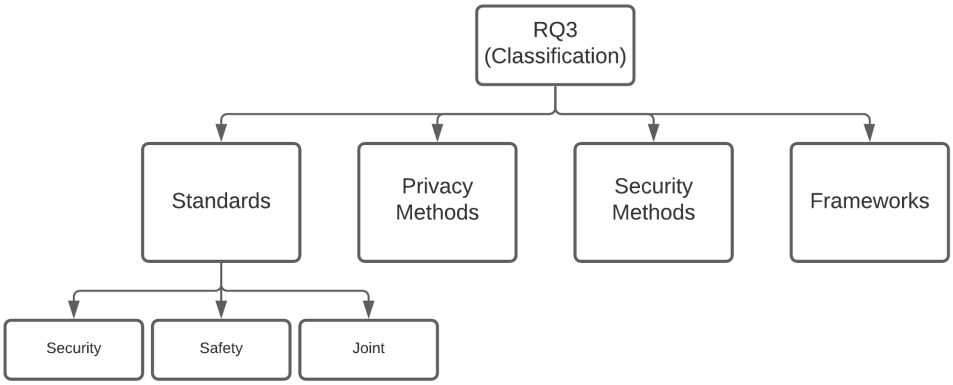


Figure C.3: Nvivo classifications for RQ3

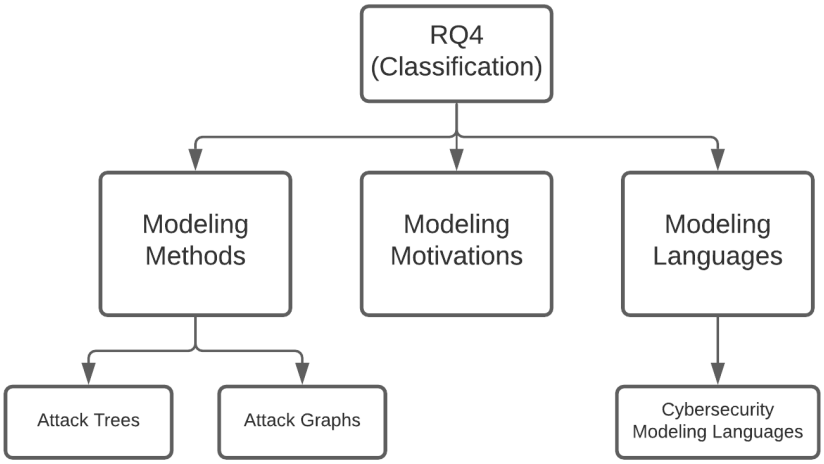


Figure C.4: Nvivo classifications for RQ4

Appendix D

Cross Cutting Examples

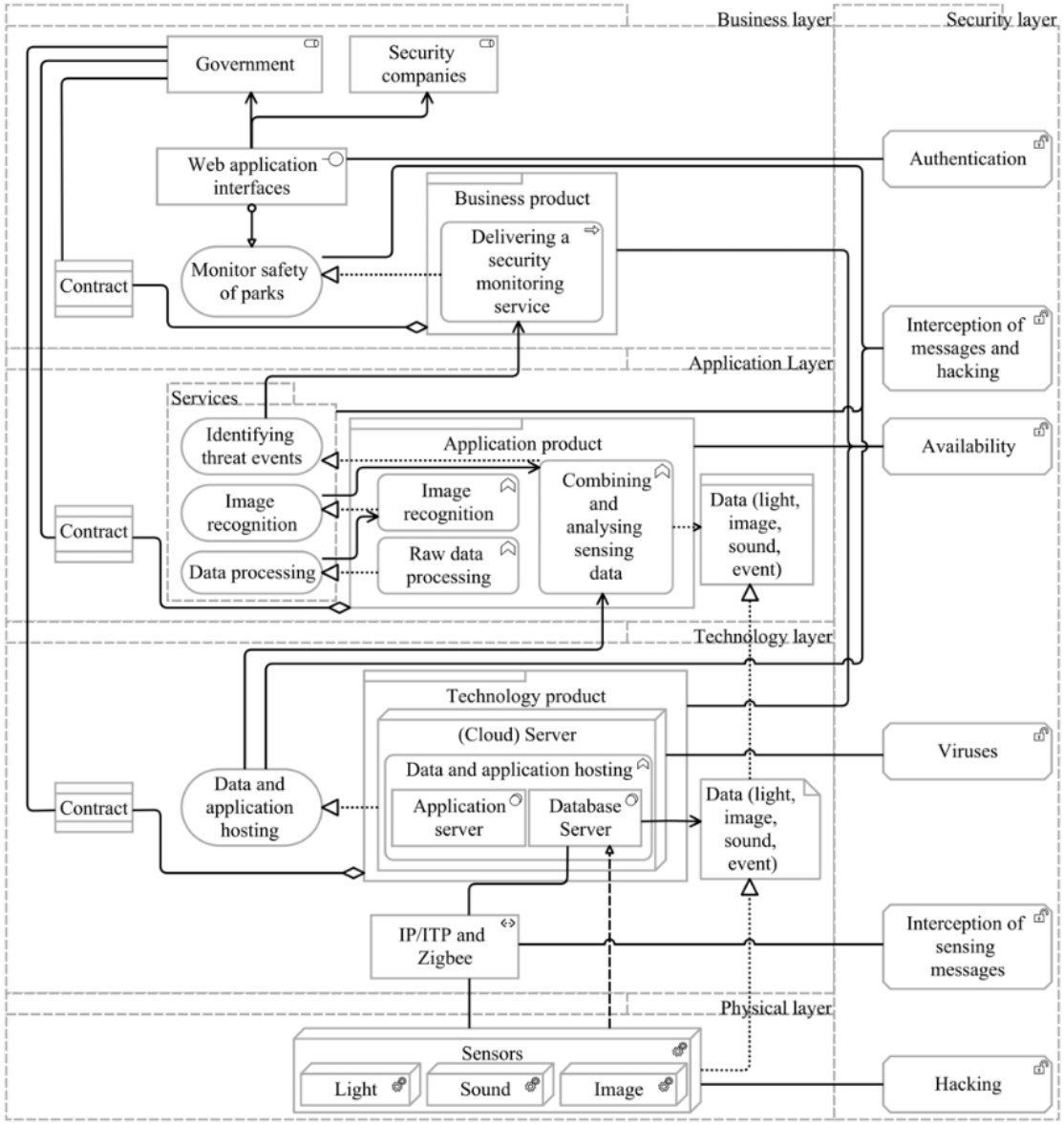


Figure D.1: Cross-cutting security architectural driver in a smart city architecture (extracted from [165])

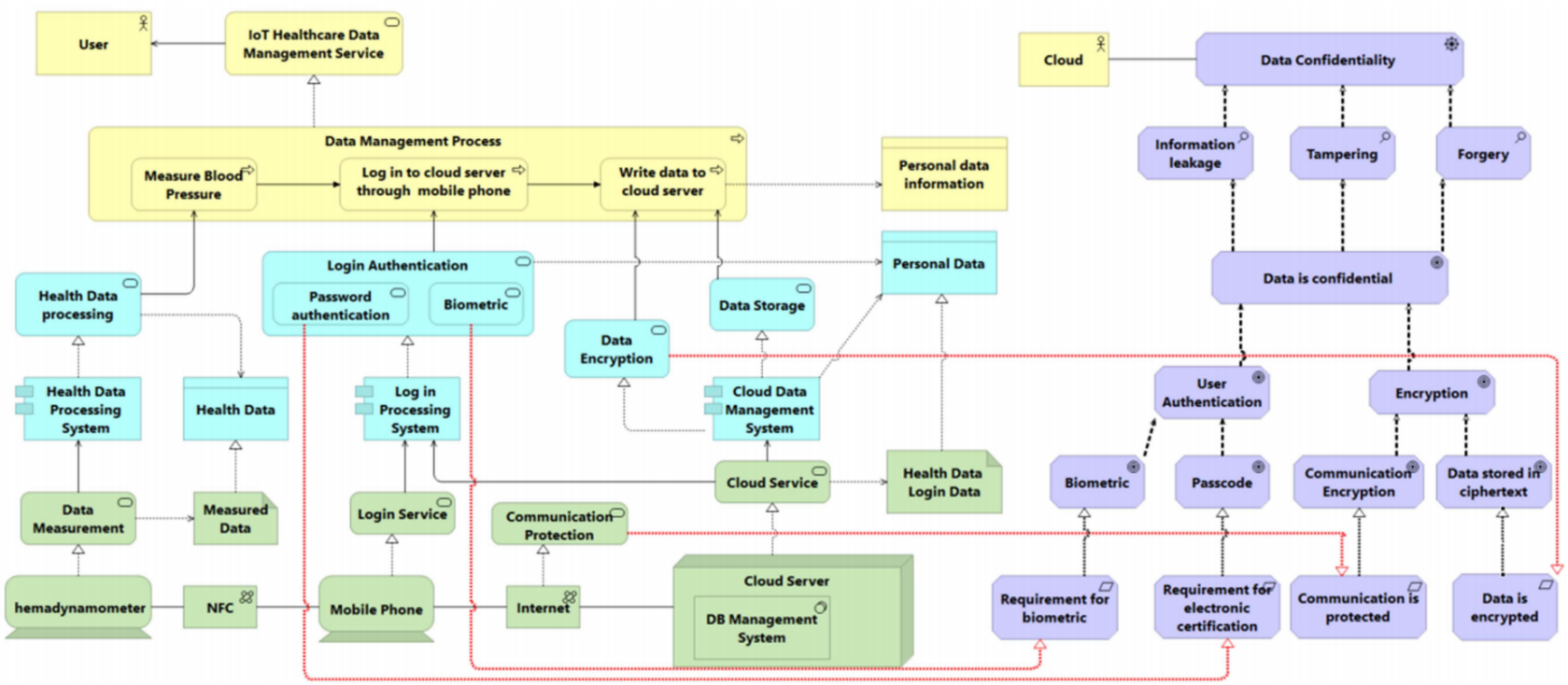


Figure D.2: Cross-cutting assurance architectural driver example (extracted from [169])

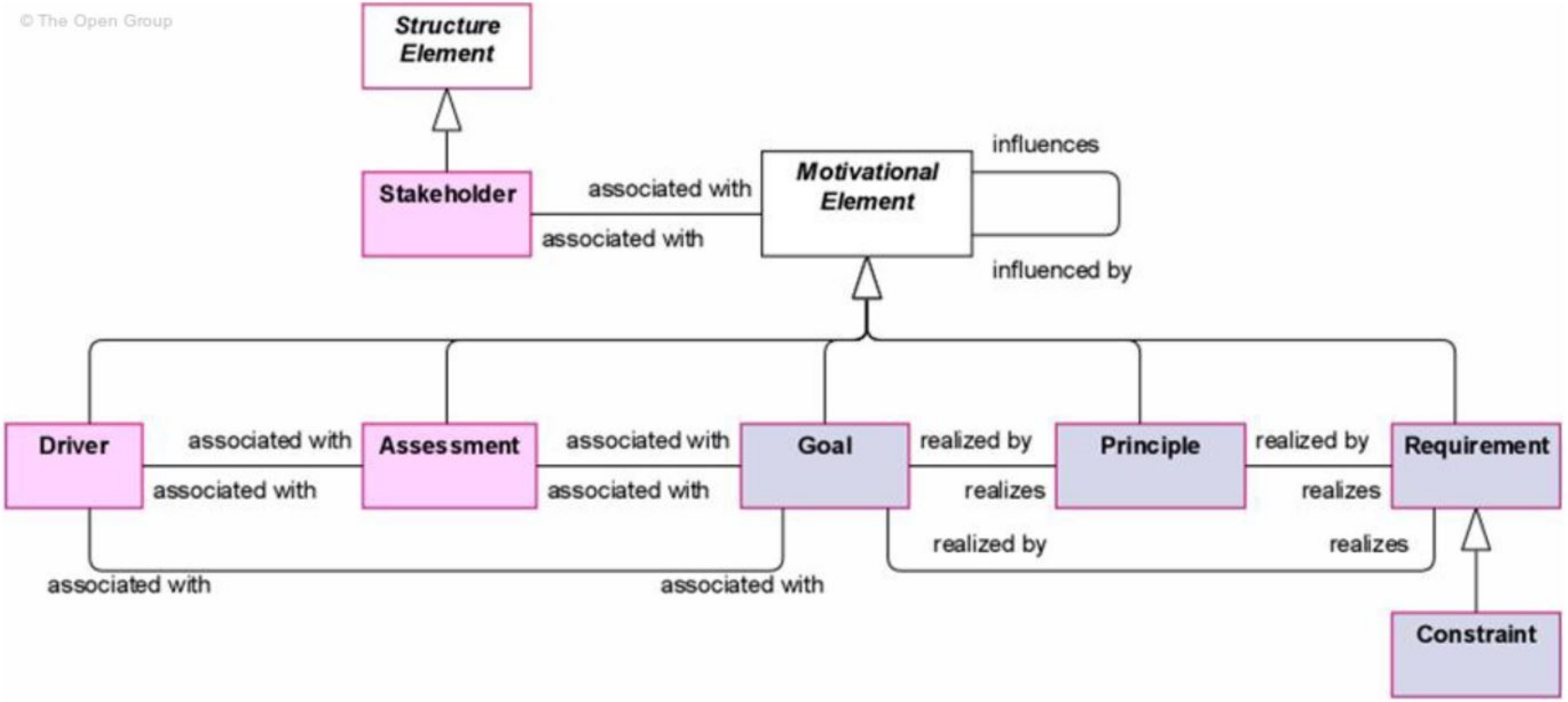


Figure D.3: Motivation aspect from ArchiMate framework (extracted from [2])

Appendix E

Attack Graph Abstraction

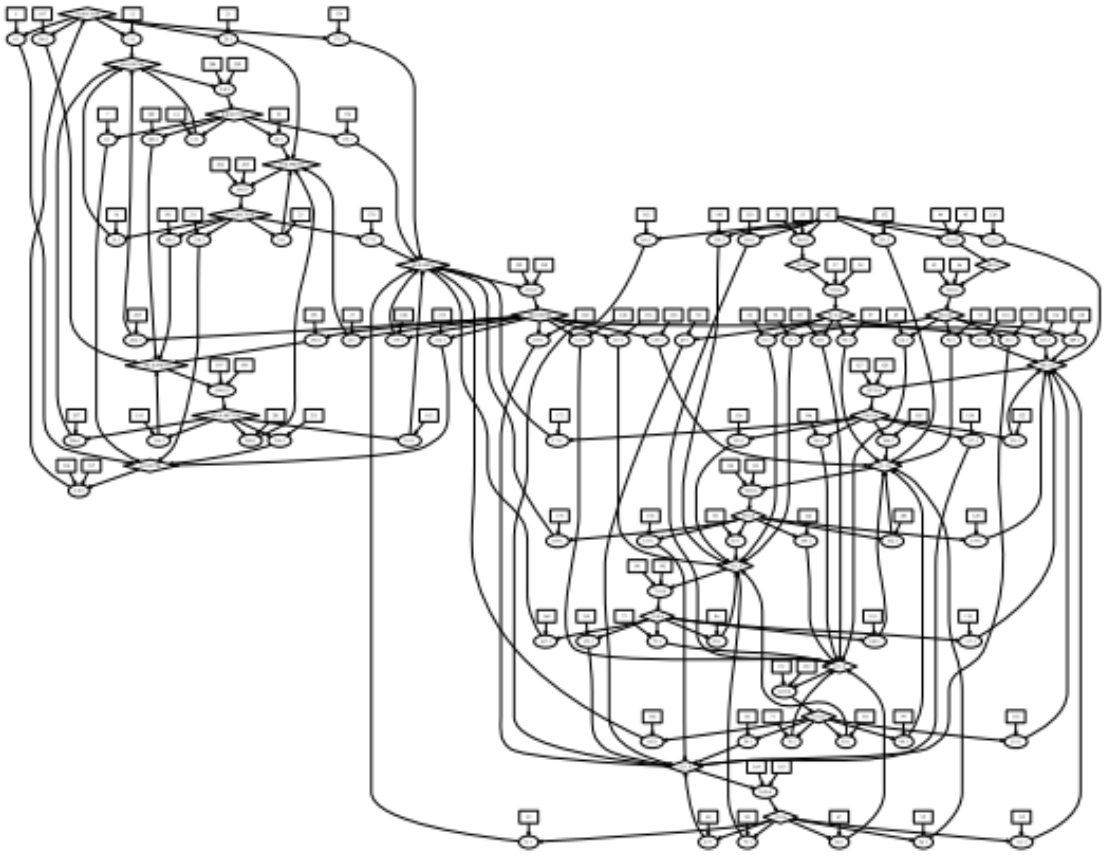


Figure E.1: Attack graph before abstraction (extracted from [177])

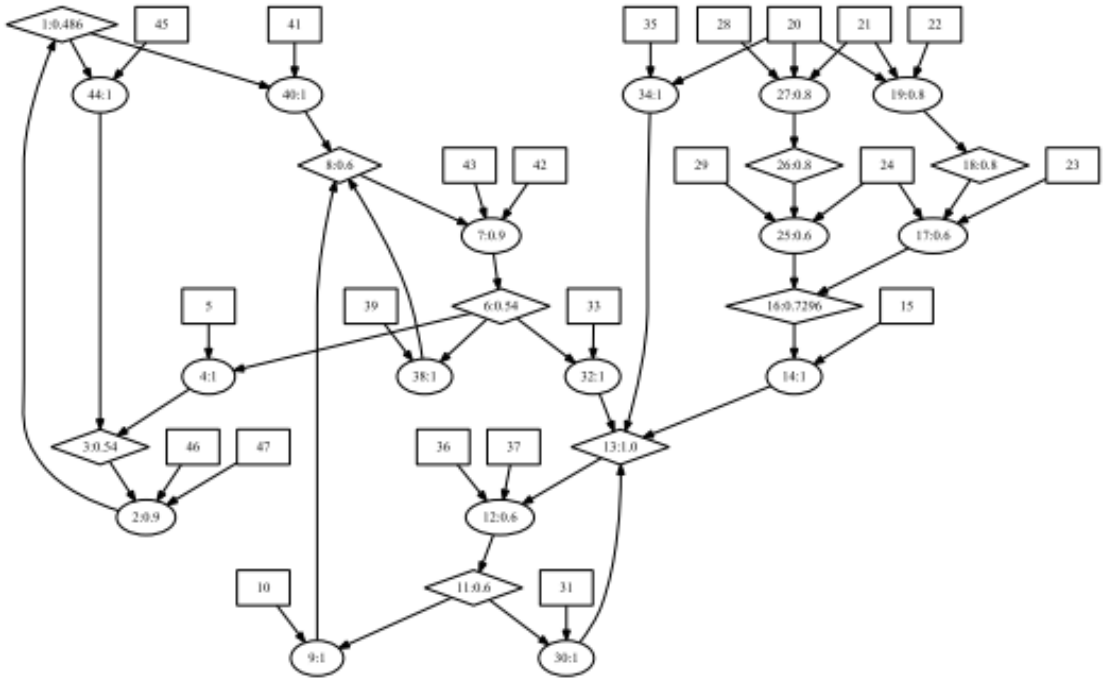


Figure E.2: Attack graph after abstraction (extracted from [177])